

Escuela Politécnica Superior

20  
21

# Trabajo fin de grado

Estimación, predicción y visualización de la demanda del servicio Acciona Movilidad



Antonio Javier Casado Hernández

Escuela Politécnica Superior  
Universidad Autónoma de Madrid  
C/ Francisco Tomás y Valiente nº 11



**UNIVERSIDAD AUTÓNOMA DE MADRID  
ESCUELA POLITÉCNICA SUPERIOR**



**Grado en Ingeniería Informática**

**TRABAJO FIN DE GRADO**

**Estimación, predicción y visualización de la  
demanda del servicio Acciona Movilidad**

**Autor: Antonio Javier Casado Hernández**

**Tutor: Daniel Táboas Rodríguez**

**Ponente: Alejandro Bellogín Kouki**

**junio 2021**

**Todos los derechos reservados.**

Queda prohibida, salvo excepción prevista en la Ley, cualquier forma de reproducción, distribución comunicación pública y transformación de esta obra sin contar con la autorización de los titulares de la propiedad intelectual.

La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual (*arts. 270 y sgts. del Código Penal*).

**DERECHOS RESERVADOS**

© 3 de Noviembre de 2017 por UNIVERSIDAD AUTÓNOMA DE MADRID  
Francisco Tomás y Valiente, nº 1  
Madrid, 28049  
Spain

**Antonio Javier Casado Hernández**

**Estimación, predicción y visualización de la demanda del servicio Acciona Movilidad**

**Antonio Javier Casado Hernández**

C\ Francisco Tomás y Valiente Nº 11

IMPRESO EN ESPAÑA – PRINTED IN SPAIN



# PREFACIO

---

Este trabajo ha sido realizado en colaboración con Acciona Movilidad con el fin de proporcionar información ante el problema de la oferta y demanda del servicio de vehículo compartido *motosharing* y mejorar las decisiones operativas relacionadas. Todo el trabajo ha sido realizado sobre datos reales del servicio.

Así mismo, el análisis de datos y patrones de movilidad que se presentan en este trabajo se realizó con soporte técnico del equipo del departamento de Transporte y Movilidad de Nommon Solutions and Technologies.

Antonio Javier Casado Hernández



# RESUMEN

---

Existen diferentes métodos de movilidad urbana alternativos al transporte con vehículo particular como son los servicios de vehículos compartido. Este tipo de servicios consiste en adquirir un vehículo estacionado en un lugar público de la ciudad y desplazarse al punto deseado. Una vez finalizado el viaje, se puede dejar el vehículo sin apenas restricciones por parte del operador del servicio de dónde ni cuando debe ser estacionado el medio de transporte.

Debido a que estos vehículos pueden ser aparcados en cualquier lugar, aunque dentro de una zona especificada por el operador, existe un reto para dar calidad al servicio ya que depende en gran medida de la disponibilidad inmediata de un vehículo en las proximidades de un usuario cuando este necesite realizar un viaje, por eso, este trabajo consiste en analizar los datos obtenidos por el uso del servicio de motocicleta compartida de Acciona Movilidad para dar información sobre zonas de alta o baja demanda y ayudar a tomar decisiones ante estos retos.

Se ha estudiado datos geoespaciales y temporales, se ha creado diferentes recursos como mapas interactivos y analizado la actividad de zonas de alta demanda para proveer de información sobre el problema de la distribución de la flota. El resultado de este trabajo da visibilidad a las zonas de alta o baja actividad y mejora la toma de decisiones operativas a la hora de distribuir y desplegar vehículos en zonas que sean necesarias.

# PALABRAS CLAVE

---

Análisis de datos, vehículo compartido, demanda constreñida, mapa, geoespacial, series temporales



# ABSTRACT

---

There are different ways of moving across cities that become alternatives to (private) transport with particular vehicles, such as shared mobility services. These services consist on obtaining a nearby parked vehicle somewhere in a public place and start a trip to the desired location. Once the trip is finished, the vehicle can be parked again with barely no restrictions by the service provider about where or when should the transport be parked.

Because these vehicles can be parked anywhere inside the allowed zone, it creates a challenge on the service to provide quality because it relies heavily on the immediate vehicle availability in the user's proximity where the user needs to start a trip. This is why this work aims to analyze data supplied by the shared vehicle service Acciona Mobility to provide knowledge about those places of high or low demand and improving the decision-making to cope with these challenges.

For this work we have studied geospatial and temporal data, we have created different resources like interactive maps and analyzed those high demand places to provide information about the problem of fleet distribution. The outcome of this work gives visibility to those high or low demand places and helps the operative decision-making related with distribute and deploy vehicles in those areas that are necessary.

# KEYWORDS

---

Data analytics, shared vehicle, constrained demand, map, geospatial, time series



# ÍNDICE

---

<b>1</b>	<b>Introducción</b>	<b>1</b>
1.1	Motivación del proyecto .....	1
1.2	Objetivos .....	2
1.3	Estructura del trabajo .....	2
<b>2</b>	<b>Estado del arte</b>	<b>3</b>
2.1	Servicios de vehículo compartido .....	3
2.2	Análisis de datos espacio temporales .....	4
2.3	Visualización de datos .....	9
2.4	Tecnologías utilizadas .....	10
<b>3</b>	<b>Diseño</b>	<b>13</b>
3.1	Punto de partida .....	13
3.2	Soluciones y objetivos .....	14
3.3	Arquitectura .....	18
3.4	Requisitos .....	19
<b>4</b>	<b>Implementación</b>	<b>21</b>
4.1	Cálculo y visualización de la demanda mediante mapas .....	21
4.2	Detección de demanda constreñida .....	22
<b>5</b>	<b>Pruebas y resultados</b>	<b>33</b>
5.1	Pruebas .....	33
5.2	Resultados .....	35
<b>6</b>	<b>Conclusión y trabajo futuro</b>	<b>39</b>
6.1	Conclusión .....	39
6.2	Trabajo futuro .....	40
	<b>Bibliografía</b>	<b>42</b>
	<b>Apéndices</b>	<b>43</b>
<b>A</b>	<b>Entornos</b>	<b>45</b>
<b>B</b>	<b>Bases de datos</b>	<b>47</b>
B.1	Descripción de los campos de las tablas de BigQuery .....	47
B.2	Consultas para la generación de datos para las series temporales .....	47
<b>C</b>	<b>Patrones de viajes en finales de lectivos</b>	<b>49</b>

C.1 Patrones de picos de viajes en finales de días no lectivos .....	49
----------------------------------------------------------------------	----



# LISTAS

---

## Lista de códigos

4.1	Filtrado de valores con denominadores a 0 y zonas de no interés .....	22
4.2	Aplicación de función logarítmica y conversión de valores infinitos .....	23
4.3	Calculo y filtro de cuadrículas dentro de la <i>geofence</i> .....	24
4.4	Consulta de viajes salientes en Atocha .....	26
4.5	Consulta de viajes entrantes en Atocha .....	26
4.6	<i>Ordinary least squares</i> y <i>fit</i> de la primera iteración del <i>Backwards stepwise regression</i> .....	31
B.1	Consulta para obtener los vehículos disponibles en Atocha .....	48

## Lista de ecuaciones

2.1	Definición de modelo AR .....	6
2.2	Definición de modelo MA .....	7
2.3	Definición de modelo ARMA .....	7
3.1	Viajes por cada <i>grid</i> normalizado .....	16
3.2	Vehículos por cada <i>grid</i> normalizado .....	16
3.3	Ratio de estimación de la demanda en un <i>grid</i> .....	16

## Lista de figuras

2.1	Ejemplo de función de autocorrelación .....	5
2.2	Ejemplo de análisis de densidad espectral .....	6
2.3	Ejemplo estacionariedad .....	7
2.4	Ejemplos de mapas de calor .....	9
2.5	Ejemplo de mapa cloropético .....	10
3.1	Mapa de tipo <i>grid</i> implementado .....	14
3.2	Flujo de trabajo para detección de demanda constreñida .....	17
3.3	Diagrama de alto nivel de la arquitectura del dashboard .....	18
3.4	Diagrama entidad relación de MySQL .....	18
3.5	Campos de las tablas de BigQuery .....	19

4.1	Cuadrícula de Madrid con mas viajes realizados . . . . .	23
4.2	Cuadrícula de Madrid con los <i>grids</i> identificados . . . . .	25
4.3	Serie temporal de viajes y vehículos disponibles de Atocha . . . . .	27
4.4	Patrón de pico de viajes en Atocha . . . . .	28
4.5	Gráfico en el que se muestra la cantidad de viajes y vehículos disponibles en Atocha los días y hoas de interés . . . . .	30
4.6	Primera iteración de <i>Backwards stepwise regression</i> . . . . .	31
5.1	Resultado de aplicar escala logarítmica . . . . .	33
5.2	Gráfico con todos los datos de viajes y vehículos de Atocha . . . . .	34
5.3	Gráfico en el que se representa cuándo se produce inconsistencias de viajes y vehículos . . . . .	35
5.4	Porcentaje de días de la semana en los que hay indicador de demanda constreñida . . . . .	35
5.5	Resultado del mapa de estimación de demanda basado en viajes y vehículos . . . . .	37
5.6	Última iteración de <i>Backwards stepwise regression</i> . . . . .	37
5.7	Predicción del modelo . . . . .	38
C.1	Pico de viajes el 13 de octubre . . . . .	49
C.2	Pico de viajes el 3 y 10 de Noviembre de 2020 . . . . .	50
C.3	Pico de viajes el 9 de diciembre de 2020 . . . . .	50

## Lista de tablas

4.1	Resultado tras consultar la cuadrícula número 439 de Madrid . . . . .	25
4.2	Tabla calendario resultante . . . . .	26

# INTRODUCCIÓN

---

El transporte es un proceso de gran importancia en todos los ámbitos, solo en la comunidad de Madrid se estimó que existían mas de 4,2 millones vehículos entre turismos, motocicletas y autobuses [1], la falta de espacio para el estacionamiento y la contaminación sobre todo en las grandes ciudades hacen que aparezcan restricciones para desplazarse con vehículo propio en estas urbes, esto junto el avance de la tecnología y las comunicaciones hacen que aparezca un nuevo método de transporte diferente al transporte privado y al transporte público: el transporte en vehículo compartido.

La libertad para poder estacionar los vehículos de este tipo de servicios en cualquier punto dentro de la zona permitida y a cualquier hora presenta problemas de oferta de vehículos ante la demanda de los usuarios. Este problema oferta-demanda es aun mas notable cuando el tipo de vehículo es una motocicleta porque puede ser estacionado con todavía menos restricciones que los turismos.

Este trabajo se centra en el análisis de los datos de Acciona Movilidad, que ofrece servicio de motocicleta eléctrica compartida. Este servicio denominado *motosharing* se basa en la posibilidad del cliente de acceder a un vehículo a través del dispositivo móvil.

## 1.1. Motivación del proyecto

Las compañías de movilidad compartida dan servicio a sus clientes de forma que cuando uno de estos usuarios quiere iniciar un viaje para desplazarse, hace uso de su dispositivo móvil y a través de una aplicación busca un vehículo cercano. Una vez localizado y habiéndose desplazado al vehículo, el cliente puede iniciar el viaje y dependiendo del tipo de servicio, tiene la posibilidad de finalizarlo en cualquier punto de una zona permitida o en una estación indicada explícitamente para dejar el vehículo.

Acciona Movilidad ofrece servicio que da libertad a la hora de iniciar un viaje en cualquier punto en el que haya un vehículo así como finalizarlo y estacionarlo en cualquier punto dentro de la zona permitida y siempre que sea un lugar legal para el estacionamiento. Esta flexibilidad para empezar y finalizar un viaje en cualquier punto puede crear problemas de desbalance de disponibilidad de vehículos en puntos de alta demanda haciendo que clientes que necesiten utilizar el servicio puedan quedar sin disponibilidad de un vehículo en las proximidades.

Existen diversas compañías que ofrecen estos servicios, sobre todo en las grandes ciudades. Los vehículos mas utilizados por estas empresas son coches, motocicletas, patinetes eléctricos y bicicletas. La competencia existente entre estas compañías hace que cada una quiera perfeccionar la disponibilidad de vehículos en zonas de demanda: un cliente que quiera realizar un viaje y no disponga de un vehículo cercano, utilizará el servicio de otra compañía que sí se encuentre en las proximidades.

Tener información sobre aquellas zonas en las que se produce desbalance de flota y cuántos posibles viajes se están perdiendo debido a poca disponibilidad de vehículos es útil para tomar decisiones operativas relacionadas con la reubicación y despliegue de la flota por parte del operador para minimizar este problema.

## 1.2. Objetivos

Este trabajo tiene como objetivo determinar aquellas zonas en las que se está produciendo déficit o superávit de vehículos en función de la demanda así como predecir cuándo va a ocurrir estos problemas y poner esta información en disposición del equipo de operaciones de la compañía mediante una aplicación web que se dispone, dedicada a la visualización de la actividad del servicio.

## 1.3. Estructura del trabajo

La estructura de este trabajo se organiza en los siguientes capítulos:

- 1.– **Introducción:** se describe brevemente los servicios de vehículo compartido, los problemas que pueden ocasionar y la motivación junto con los objetivos para resolverlos.
- 2.– **Estado del arte:** se explica en detalle los servicios de vehículo compartido así como también el servicio de Acciona Movilidad. Se describe conceptos y modelos para análisis de series temporales, sistemas de información geográfica, tecnologías para visualización de datos y por último una breve descripción de las tecnologías usadas y para qué se han utilizado en el trabajo.
- 3.– **Diseño:** incluye el punto de partida, la arquitectura del sistema, el diseño de las soluciones propuestas y los requisitos
- 4.– **Implementación:** se describe la implementación del diseño, posibles problemas y resultados más significativos
- 5.– **Pruebas y resultados:** se expone algunas pruebas realizadas para el trabajo y se interpreta los resultados obtenidos en la implementación.
- 6.– **Conclusión y trabajo futuro:** resumen del trabajo y sus resultados junto con posibles tareas futuras que no se ha abarcado y que podría extender las soluciones que este proyecto presenta.

## ESTADO DEL ARTE

---

En este capítulo se exponen los tipos de servicios de vehículo compartido junto con las diferentes tecnologías y metodologías con las que se analiza los datos geoespaciales y temporales para resolver diversos problemas relacionados con estos servicios.

### 2.1. Servicios de vehículo compartido

Los servicios de movilidad compartida son aquellos en los que los viajeros comparten un vehículo a lo largo del tiempo como alquiler personal (a diferencia del transporte público) y compartiendo el coste del viaje. Los vehículos de este tipo de servicios pueden constar de coches, motocicleta u otro tipo de vehículos como bicicletas o patinetes eléctricos.

#### 2.1.1. Tipos de servicio de movilidad compartida

Existen diferentes tipos de servicio de vehículo compartido que son de interés para el estudio de su optimización.

##### Basado en estaciones

La libertad para empezar y finalizar un viaje en este tipo de servicio está restringido a localizaciones específicas en las que se apilan los vehículos, en este caso es necesario planificar el viaje para comprobar la disponibilidad de las estaciones. Se han hecho estudios en los que se concluyen qué estaciones son demasiado demandadas y cuáles no, en cuáles predominan los viajes entrantes y cuáles predominan los salientes [2].

##### Sin restricciones para la recogida y aparcamiento

Denominado *Free-floating car sharing* (FFCS) [3], los usuarios pueden empezar y finalizar el viaje en cualquier punto de una zona permitida. Las diferentes empresas que operan este tipo de servicios

funcionan de forma similar: el cliente que quiere desplazarse y se encuentra dentro de la zona permitida del servicio hace uso de su teléfono móvil, a través de una aplicación, para buscar un vehículo cercano disponible, el cliente se desplaza al vehículo y lo desbloquea a través de la aplicación, entonces inicia el viaje a su destino y lo estaciona en una zona permitida, por último, finaliza el viaje a través del dispositivo móvil. El coste monetario cargado al cliente depende del tiempo de su viaje. Esto introduce la necesidad operativa de balancear la flota de acuerdo a la demanda en una determinada zona.

Existe una extensa literatura para optimización de reubicaciones para servicios de vehículo compartido basado en estaciones [4], sin embargo, existe una brecha en el estudio del balanceo de los sistemas FFCS debido a la dificultad para obtener los datos de las compañías que los operan [5].

### **2.1.2. Acciona Movilidad**

El servicio de Acciona Movilidad que se estudia en este trabajo no tiene restricciones para la recogida o aparcamiento, es decir, es de tipo FFCS, con la característica de que su flota se compone de motocicletas eléctricas, dando todavía mas libertad a los clientes para finalizar su trayecto en cualquier punto de la zona permitida. Aunque la normativa depende de cada municipio, estos vehículos se pueden aparcar por ejemplo en las aceras sin necesitar zonas de aparcamientos expresas como en el caso de los turismos.

La compañía opera actualmente en 8 regiones distintas, cada región se compone de una o varias áreas de operaciones formadas por geocercas o *geofences* del inglés, así mismo, una región puede abarcar varias localidades próximas y es posible desplazarse entre ellas. En este trabajo, el estudio se centrará en Madrid aunque los resultados y conclusiones extraídos se puede aplicar a cualquier región definida por la compañía.

## **2.2. Análisis de datos espacio temporales**

Los datos espacio temporales son usados en análisis de datos cuando la información es recogida a través de ambos espacio y tiempo, describiendo así fenómenos en un cierto lugar y tiempo por ejemplo un vehículo que se desplaza por una zona podría registrar su posición a cada intervalo de tiempo.

A continuación se presentan los conceptos, algoritmos y tecnologías para analizar datos con información de espacio y tiempo.

### **2.2.1. Series temporales**

Una serie temporal es una secuencia discreta de puntos de datos ordenados en el tiempo y que se pueden agrupar en rangos iguales o desiguales de tiempo para sacar datos estadísticos significativos.

Con las series temporales se pueden predecir valores futuros mediante un modelo basado en valores ya observados. Algunas características importantes de las series temporales son las siguientes:

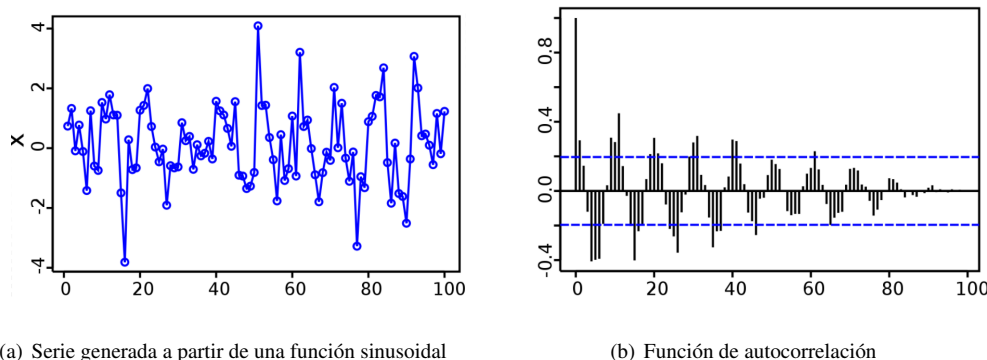
### Estacionalidad

En muchas ocasiones las series temporales reflejan variaciones de valores que ocurren a intervalos regulares, es decir, tienen un comportamiento periódico.

**Autocorrelación:** Una función de autocorrelación, *autocorrelation function* del inglés (ACF), revela la forma en la que la correlación entre dos valores de una señal o serie temporal cambia a medida que su separación cambia [6], es decir, es la similitud de observaciones como función de tiempo de separación entre ellas. Un análisis de la autocorrelación sirve para encontrar patrones de una función y que en ocasiones pueden estar escondidos por valores aleatorios llamados ruido.

Una función de autocorrelación tiene forma como la de la figura 2.1(b) en la que se puede observar una función sinusoidal que disminuye a medida que aumenta su separación, lo que significa que existe una estacionalidad en su serie, representada en la figura 2.1(a).

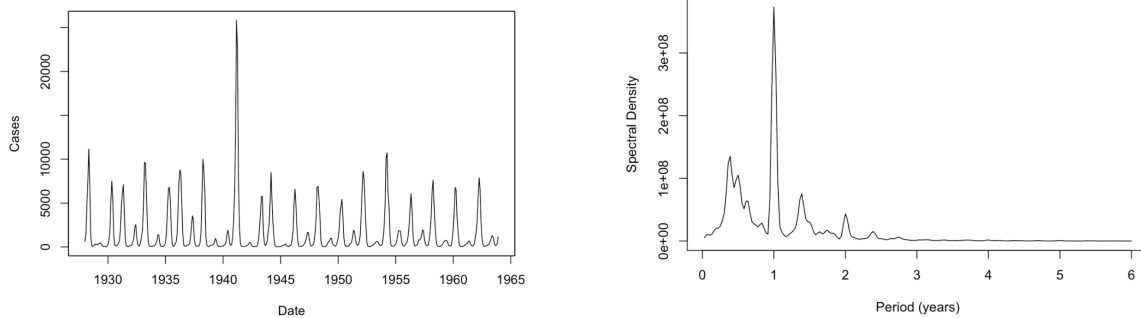
**Autocorrelación parcial:** Es similar a la función de autocorrelación, pero eliminando las relaciones de las observaciones intermedias. Por ejemplo, una función de autocorrelación de orden 3 calcula la correlación entre la serie temporal en los instantes  $t_1, t_2, t_3 \dots t_n$  y la propia serie en los instantes  $t_4, t_5, t_6 \dots t_n$  pero eliminando los efectos de las separaciones 1 y 2.



**Figura 2.1:** Serie temporal 2.1(a) y su función de autocorrelación 2.1(b)

**Análisis de densidad espectral:** Al igual que en ACF, esta es una técnica que permite descubrir periodicidades de una función pero en este caso permite revelar la frecuencia a la que ocurre.

Un ejemplo de análisis espectral es la estimación de la periodicidad de casos de sarampión: en la figura 2.2(b) se observa un pico en el periodo de 1 año, indicando periodicidad cada este intervalo de tiempo.



(a) Serie temporal de los casos de sarampión

(b) Densidad espectral respecto al periodo en los casos de sarampión

**Figura 2.2:** Análisis espectral de los casos de sarampión [7]. La figura 2.2(a) representa los casos respecto al tiempo, en la figura 2.2(b) muestra la densidad espectral respecto al periodo. Se puede observar como el periodo de casos se repite fuertemente cada año.

### Series estacionarias

Una serie temporal es estacionaria cuando las propiedades estadísticas del proceso que la genera, llamado proceso estocástico, no cambian a lo largo del tiempo. Esto no significa que la serie no cambie con el tiempo, si no que la forma con la que cambia no cambia a lo largo del tiempo.

Una serie es estacionaria si cumple que :

- Tiene media constante.
- Tiene desviación típica constante.
- No hay estacionalidad

Un ejemplo de serie estacionaria se muestra en la figura 2.3(a) y una serie no estacionaria en 2.3(b). La estacionariedad es importante porque las series con esta propiedad son mas fáciles de analizar ya que los parámetros del proceso que la genera permanecen constantes.

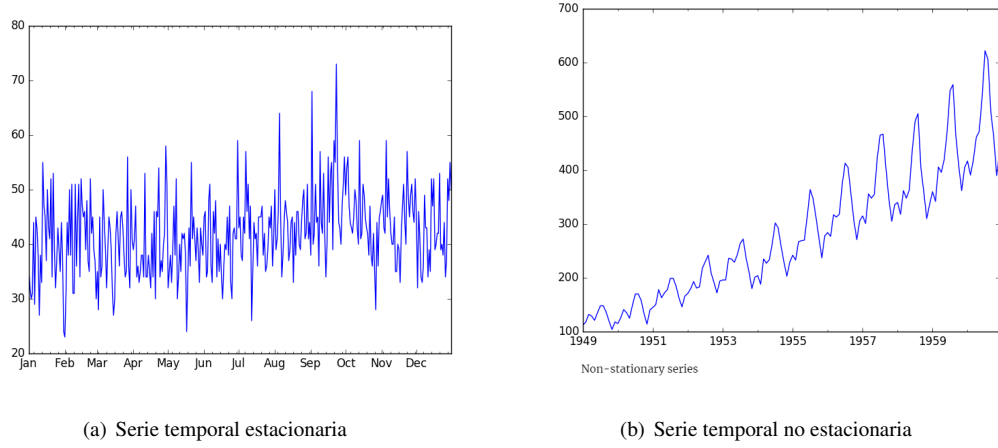
### 2.2.2. Modelos de series temporales

Las series temporales se pueden describir y predecir valores futuros mediante los siguientes modelos.

#### Modelo Auto regresivo

Del inglés *Autoregressive* (**AR**). Es un modelo que observa una regresión de una variable consigo misma desplazada.  $AR(q)$  es un modelo autoregresivo de orden  $q$  lo que indica un modelo basado en sus  $q$  valores atrasados y se define mediante la ecuación 2.1.





**Figura 2.3:** Estacionariedad y no estacionariedad en series temporales [8]. La figura 2.3(a) representa una serie temporal estacionaria, 2.3(b) muestra una serie no estacionaria

$$X_t = c + \sum_{i=1}^q \varphi_i X_{t-i} + \epsilon_t \quad (2.1)$$

Donde  $X_t$  es la observación en el instante  $t$ ,  $c$  es una constante,  $\varphi_1 \dots \varphi_q$  son los parámetros del modelo y  $\epsilon_t$  es ruido.

### Modelo Media móvil

Del inglés *Moving Average* (**MA**). Este modelo usa la dependencia entre una observación y el error residual de un modelo aplicado a observaciones previas. Su notación es  $MA(q)$  lo que indica que el modelo tendrá en cuenta los  $q$  errores pasados. Este modelo se define mediante la ecuación 2.2.

$$X_t = \mu + \sum_{i=1}^q \Theta_i \epsilon_{t-i} + \epsilon_t \quad (2.2)$$

Siendo  $\mu$  la media de la serie,  $\Theta_1 \dots \Theta_q$  los parámetros del modelo y  $\epsilon_t$ , los errores en el momento  $t$ .

### ARMA

*Autoregressive Moving Average* combina los modelos AR y MA descritos anteriormente, se denota de forma  $ARMA(p, q)$  siendo  $p$  el grado de la parte AR y  $q$  el grado de la parte MA y se define mediante la ecuación 2.3.

$$X_t = c + \sum_{i=1}^p \varphi_i X_{t-i} + \sum_{i=1}^q \Theta_i \epsilon_{t-i} + \epsilon_t \quad (2.3)$$

Al combinar AR y MA, este modelo da una descripción de una serie temporal basado en sus valores anteriores y en el error residual de un modelo aplicado a valores anteriores.

## ARIMA

*Autoregressive Integrated Moving Average*, es una generalización del modelo ARMA. ARIMA es usado en series temporales no estacionarias convirtiéndolas en estacionarias, lo que introduce la parte *Integrated*, mediante la diferencia entre los valores en un momento dado y el siguiente. Así por ejemplo un serie descrita por un proceso que va ascendiendo linealmente de la forma 2,4,8,10 ... la parte *Integrated* tendría en cuenta la diferencia de cada valor y el siguiente, en este caso 2, manteniendo así el proceso constante y la serie estacionaria.

Los parámetros de los modelos anteriormente explicados se hallan analizando las funciones de autocorrelación y autocorrelación parcial.

### 2.2.3. Sistemas de información geográfica

Los sistemas de información geográfica, *Geographic Information System* del inglés (GIS) son herramientas que permiten el recabado, almacenamiento y análisis de información espacial y geográfica. Esto permite a un usuario visualizar datos, editarlos y consultarlos de forma interactiva para representarlos en mapas [9] [10].

Los sistemas de información geográfica se utilizan usando diferentes tecnologías y en multitud de ámbitos como ingeniería, logística y transportes, investigación científica, negocios y telecomunicaciones [10]. Por eso GIS está relacionada con la llamada inteligencia de localización.

#### Inteligencia de localización

La inteligencia de localización se usa para obtener información significativa a partir de datos geoespaciales para resolver un problema concreto. El primer ejemplo documentado de inteligencia de localización fue en Londres en 1854 cuando el médico John Snow, con el fin de investigar y paliar un brote de cólera en la ciudad, fue capaz de relacionar los brotes de la enfermedad con la localización de bombas de aguas contaminadas [11] [12], para entonces no se sabía que las enfermedades eran causadas por microorganismos y los contagios de cólera se achacaban a un tipo de contaminación del aire (la obsoleta teoría miasmática de la enfermedad) [13]. Aunque John Snow no fue capaz de explicar por qué ese agua estaba causando los brotes, sus investigaciones de los patrones de los contagios convenció a la población local para no hacer uso de esas bombas de agua.

En la actualidad, los sistemas de información geográfica y la inteligencia de localización se realizan mediante *software* permitiendo almacenar grandes cantidades de datos y ejecutar sobre ellos algoritmos de forma eficaz y rápida.

### 2.2.4. Sistemas de referencia geográfica

Los sistemas de referencia geográfica sirven para ubicar un elemento sobre la superficie de la tierra, aunque existen varios sistemas de coordenadas geográficas, en este trabajo se usarán los sistemas de coordenadas latitud y longitud:

- Latitud: indica la posición norte-sur sobre la superficie de la tierra.
- Longitud: indica la posición este-oeste sobre la superficie de la tierra.

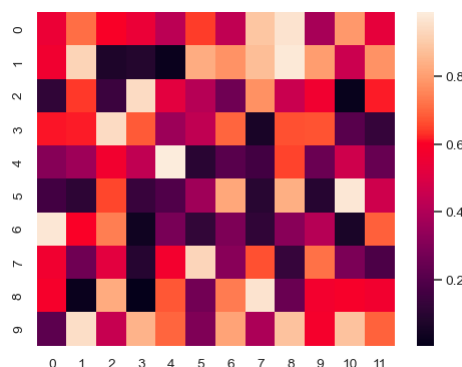
Estos sistemas de coordenadas son comúnmente utilizados con grados sexagesimales (grados, minutos y segundos) y con grados decimales. Este último es el tipo en el que está almacenado en las bases de datos y el recomendado para el uso en sistemas digitales [14] por lo será el que se use en este trabajo.

## 2.3. Visualización de datos

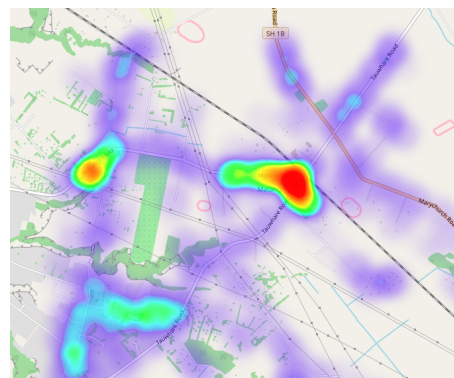
A continuación se muestran algunos métodos y tecnologías comúnmente utilizados para la visualización de datos geoespaciales.

### Mapa de calor

Un mapa de calor es una forma de visualización de datos que muestra una magnitud de interés a través del color. Esto se puede representar en una matriz, representando cada elemento de esta con un color tal y como se muestra en la figura 2.4(a) o espacial en la que se representa la magnitud también por color pero una posición determinada, un ejemplo es el de la figura 2.4(b).



(a) Mapa de calor en matriz

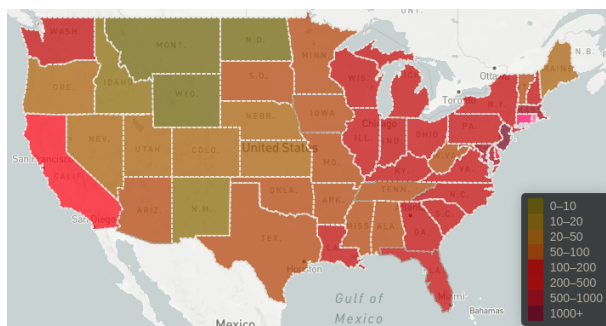


(b) Mapa de calor espacial

**Figura 2.4:** Ejemplos de mapas de calor de tipo matriz [15] en el que el color indica el valor de la posición y de tipo espacial [16]

## Choropleth

Un mapa coroplético o *choropleth* del inglés es un tipo de mapa espacial en el que se representa sobre ellos valores estadísticos de una determinada región a través de colores, un ejemplo de este tipo de mapa se muestra en la figura 2.5. Las regiones o distritos en ese tipo de mapas son predefinidas para luego calcular sobre ellos alguna variable que defina esa región.



**Figura 2.5:** Mapa de tipo coroplético [17], el color de cada región indica la densidad de población en esta.

## 2.4. Tecnologías utilizadas

Las tecnologías usadas en este trabajo han sido consideradas de acuerdo a los requisitos funcionales y no funcionales, detallados en la sección 3.4.

### 2.4.1. Bases de datos

Todos los datos analizados han sido obtenidos mediante consultas a gestores de bases de datos.

- **Google BigQuery:** es un gestor de bases de datos de Google Cloud capacitado para analizar cantidades de datos de orden de petabytes [18]. Es en esta base de datos donde se almacena todos los estados y posiciones de los vehículos en todo momento y tiene un coste monetario por consulta que depende de la cantidad de datos procesados.
- **MySQL:** gestor de bases de datos relacional en el que almacena información relacionada con los viajes y vehículos del servicio además de otra información menos relevante para este trabajo como datos de usuarios, facturaciones etc. A diferencia de BigQuery, en esta base de datos no se almacenan las posiciones y estados históricos de los vehículos.

### 2.4.2. Visualización de datos mediante mapas

- Backend y procesado de datos.

- **Python:** se utiliza este lenguaje de programación, es versátil y adecuado para el análisis y procesamiento de datos, además es en el que está implementado el *framework* de backend.
- **Flask:** micro *framework* para el backend de la aplicación.
- **Pandas:** librería de python para el análisis y manipulación de datos, usado en este trabajo para manejar estructuras de datos provenientes de bases de datos SQL y realizar operaciones.
- **GeoPandas:** extensión de Pandas para trabajar con datos geoespaciales, permite realizar operaciones espaciales y geométricas.
- Frontend.
  - **JavaScript:** lenguaje del lado del cliente que prepara los datos para su visualización.
  - **Leaflet:** librería de JavaScript para construir mapas interactivos en aplicaciones web y móviles [19].
  - **Material Dashboard:** diseño de frontend para la aplicación web .
- Despliegues.
  - **Docker:** usado para contener y ejecutar software en paquetes llamados contenedores, utilizado en este trabajo para realizar los despliegues en el entorno de desarrollo. En el apéndice A se detalla los entornos en los que se ha trabajado.

### 2.4.3. Experimentos y análisis de series temporales

- **Jupyter notebook:** entorno de desarrollo interactivo con python, usado para combinar código, resultados y texto.
- **Plotly:** librería de python de código abierto para construir gráficos interactivos.
- **Folium:** librería de python de código abierto para construir mapas, funciona sobre Leaflet. En este trabajo se ha preferido usar Leaflet para el *dashboard* debido a que los mapas generados por Folium son estáticos y sin posibilidad de añadir funcionalidad a estos.
- **Statsmodel:** librería de python para la obtención y estimación de modelos estadísticos [20]. Usado en este trabajo para realizar regresión para el modelo de serie temporal.



## DISEÑO

---

### 3.1. Punto de partida

Este trabajo se realiza como continuación de unas prácticas en empresa en la que se ha aportado mejora y desarrollo de una aplicación web, llamado *dashboard*, que muestra una visión general del servicio. Los usuarios de esta aplicación son los empleados internos de Acciona Movilidad. La aplicación contiene un apartado de operaciones que genera información de la actividad de la flota de vehículos a partir de unas opciones que el usuario puede elegir.

El *dashboard* muestra unos formularios que permiten elegir la región a mostrar y el rango de fecha y hora en el que se quiere agrupar los datos. Dependiendo del tipo de dato a analizar, la aplicación genera un tipo de mapa junto con una determinada información. Algunas de estas opciones son:

#### Mapas generales

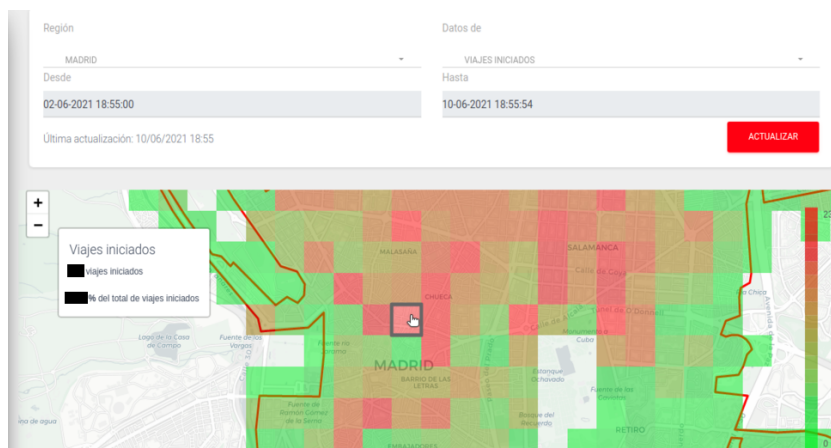
Generan un mapa con distinta información dependiendo de la opción a elegir, algunas de estas son:

- Mapa de vehículos parados: muestra sobre un mapa todos los vehículos del servicio de la ciudad indicada sobre un mapa de la región junto con alguna información de cada uno, por ejemplo el estado del vehículo, su matrícula, último viaje realizado etc.
- Mapa de agrupación de vehículos: muestra los puntos de la región indicada en la que se producen agrupaciones o *clusters* de vehículos aparcados.

#### Mapas de tipo *Grid*

Genera un mapa de tipo coroplético que divide la región seleccionada en regiones iguales llamados *grids* y cada una de estas regiones agrupan información de un dato elegido por el usuario, cada *grid* es representado mediante un color dependiendo del valor numérico del dato. En la figura 3.1 se muestra este tipo de mapa. Algunas de las opciones de dato a mostrar son:

- Distribución de flota: Muestra información del número de vehículos actuales aparcados en cada *grid*.
- Tiempo medio de parada: Indica el tiempo medio que llevan disponibles los vehículos que se encuentran dentro



**Figura 3.1:** Mapa *grid* ya implementado en el que se muestra en cada zona seleccionada el número de viajes que ha habido en el rango de tiempo seleccionado en el formulario superior (se ha censurado el detalle exacto del dato por motivos de confidencialidad).

de cada *grid* para realizar un viaje por los usuarios.

- Viajes iniciados: Muestra el número de viajes iniciados en cada *grid* en una región y un rango de fechas a elegir por el usuario.

Una parte de este trabajo se realiza añadiendo funcionalidad a este *dashboard* ya implementado, lo que genera la mayoría de requisitos funcionales y no funcionales de la Sección 3.4.

## 3.2. Soluciones y objetivos

El objetivo general de este trabajo es proporcionar conocimiento sobre la demanda y la oferta del servicio Acciona Movilidad para abarcar los problemas de déficit de disponibilidad de vehículos ante la demanda de los clientes, para ello se ha dividido el trabajo en 2 partes generales que se detallan a continuación.

### 3.2.1. Cálculo y visualización de la demanda mediante mapas

Esta parte del trabajo consiste en añadir funcionalidad al *dashboard* en la que se calculan y visualizan datos relacionados con la oferta y demanda de vehículos para aportar información de en que zonas se necesita mas disponibilidad de vehículos o en que zonas se necesitaría retirar vehículos debido a que no se producen viajes. Esto mejora la toma de decisiones del equipo de operaciones del servicio, este equipo es el que se encarga de los despliegues de vehículos, reubicaciones, cambios de batería etc.



## Cuestiones no resueltas con la funcionalidad existente

Si bien ya se dispone de algunos datos y de funcionalidad que muestran información sobre la actividad del servicio tal y como se ha explicado en la sección 3.1, esto no proporciona suficiente referencia para tomar decisiones operativas relacionadas con el problema de disponibilidad de vehículos. Los mapas ya implementados de tipo *grid* proporcionan información sobre este tema pero quedan cuestiones sin responder:

- El mapa de **distribución de flota** no provee de datos suficientes para determinar la demanda de vehículos en una zona, tan solo muestra la oferta.
- El mapa de **tiempo medio de parada** podría indicar un valor relacionado con la demanda: zonas con bajo tiempo medio de parada podría ser una zona con alta demanda, sin embargo, no se tiene en cuenta los siguientes problemas, entre otros:
  - La diferencia de vehículos disponibles en cada *grid* daría valores pocos significativos, así por ejemplo una zona con tan solo un vehículo en el que normalmente se realiza un viaje por hora, daría (como mucho) 1 hora de tiempo medio de parada, sin embargo una zona con 20 vehículos en la que se realiza 20 viajes por hora y cada uno de esos viajes se distribuyeran de forma uniforme en cada vehículo, también daría un tiempo medio de parada de 1h, sin embargo, la demanda es mucho mayor en la última.
  - Proporciona información instantánea, es decir, tan solo se tiene el dato del tiempo medio que llevan parados los vehículos que se encuentran en cada *grid* en el momento de generar el mapa pudiendo cambiar esto dependiendo de la hora del día.
  - Zonas con muy alta demanda pueden también disponer de alta oferta (debido a viajes que finalizan y dejan el vehículo disponible allí) y es muy probable que se cree un efecto pila por cuestiones de espacio: esto es, que se usen primero los vehículos que han llegado últimos, aumentando así el tiempo de parada de los vehículos primeros y aumentando también el tiempo medio de la zona.
  - Un vehículo averiado o con algún desperfecto también podría aumentar el tiempo medio del *grid* de forma desproporcionada.
- El mapa de **viajes iniciados** no revela que zonas están teniendo un déficit de vehículos así pues una zona con un número alto de viajes puede también estar sujeta a déficit de vehículos o de igual forma puede tener un superávit de vehículos debido a que entran mas viajes que salen.

## Solución

A continuación se idea un desarrollo que puede proporcionar información a los problemas previamente explicados aportando funcionalidad al *dashboard*.

## Estimación de demanda basada en viajes y en vehículos

Una estimación para la importancia relacionada con la oferta y la demanda de una zona es la cantidad de viajes realizados junto el número de vehículos disponibles en esa zona. Una zona con escasos vehículos en la que se ha realizado gran cantidad de viajes indicaría un problema de déficit de vehículos, de igual forma una zona con muchos vehículos y escasos viajes históricos, señalaría una

zona en la que sería necesario retirar vehículos.

Esta solución consiste en agrupar datos de viajes históricos del servicio junto con las posiciones de los vehículos actuales para detectar aquellas zonas en las que sea necesario realizar acciones operativas.

El parámetro de interés en cada zona sería una comparación entre el número de viajes que ha habido en esa zona y el número de vehículos que se encuentran allí actualmente, para ello, sería necesario estandarizar el número de viajes y de vehículos en cada zona, es decir, para cada *grid* se calcula estos parámetros:

$$T_g = \frac{viajes_g}{\sum_{g=0} viajes_g} \quad (3.1)$$

$$V_g = \frac{vehiculos_g}{\sum_{g=0} vehiculos_g} \quad (3.2)$$

Donde  $g$  es el *grid* (cuadrícula individual) en el que se calcula los parámetros. El denominador de la ecuación 3.1 refleja el número de viajes iniciados en todos los *grids* de la ciudad elegida dentro de un rango de fechas elegido por el usuario. El denominador de la ecuación 3.2 refleja el número total de vehículos actuales en la ciudad.  $T_g$  y  $V_g$  estaría comprendidos entre 0 y 1.

El motivo por el que se agrega el número de viajes en una rango de fechas elegido por el usuario, es porque la tendencia de la actividad puede cambiar a lo largo del tiempo, como por ejemplo, estaciones con lluvia tendría menor número de viajes que estaciones con buen tiempo.

Con estos dos parámetros calculados se puede obtener el ratio que estima la oferta contra la demanda:

$$D_g = \frac{T_g}{V_g} \quad (3.3)$$

En la sección 4.1.1 se detalla como se implementa está solución y como se abarca un posible denominador a cero.

Debido a que diferentes días de la semana tiene comportamiento diferente en la actividad de los usuarios, sobre todo en fin de semana, también se aplicará un filtro por días de la semana en el que se pueda elegir uno o varios días. Con este filtro será posible agregar los datos de viajes por los días seleccionados.

### 3.2.2. Detección de demanda constreñida mediante series temporales

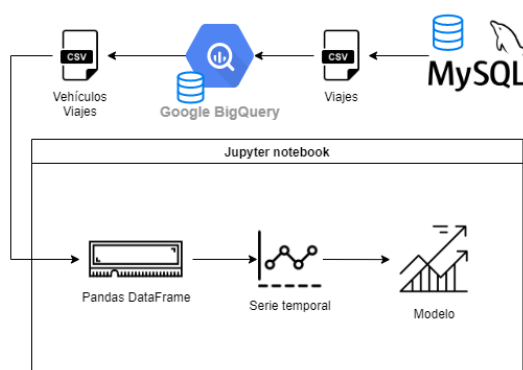
Uno de los retos para el desarrollo de modelos de demanda de movilidad compartida basados en registros históricos es que parte de las observaciones del número de viajes en el pasado están condicionadas por el número de vehículos disponibles, lo denominado como demanda constreñida. Es decir, aquellas zonas en las que se están realizando pocos viajes porque hay pocos vehículos pero que en cambio podría haber mas viajes si hubiera mas vehículos sería una zona con demanda constreñida.

La detección de zonas con este problema se realiza a nivel de *grid* y el objetivo es detectar en qué zona se están perdiendo viajes y cuantos de ellos sería posible realizarse si hubiera suficientes vehículos. Así como anticiparse a este problema.

#### Flujo de trabajo

En la figura 3.2 se muestra el flujo de trabajo para la detección de zonas de demanda constreñida, los pasos necesarios son:

- 1.– **Obtención del histórico de viajes:** se ejecuta una consulta contra MySQL para obtener los datos de viajes históricos del servicio para luego ser pasado a un archivo CSV.
- 2.– **Obtención histórico de estados de vehículos:** se obtiene los datos de los vehículos en BigQuery y será unido junto con los datos de viajes obtenidos de MySQL.
- 3.– **Exportación a CSV:** se exporta este archivo con los datos conjuntos de viajes y vehículos. La decisión de exportarlo a CSV se debe a que la consulta de estados históricos de los vehículos contra BigQuery tiene gran coste monetario por lo que una vez exportado, no será necesario realizar mas consultas, además reducirá el tiempo de lectura de datos.
- 4.– **Serie temporal:** se construye una serie temporal de vehículos y viajes leyendo los datos mediante pandas.
- 5.– **Creación de modelo:** se analiza la serie temporal y se crea un modelo para predecir caídas de viajes con antelación y cuantificar posibles pérdidas.

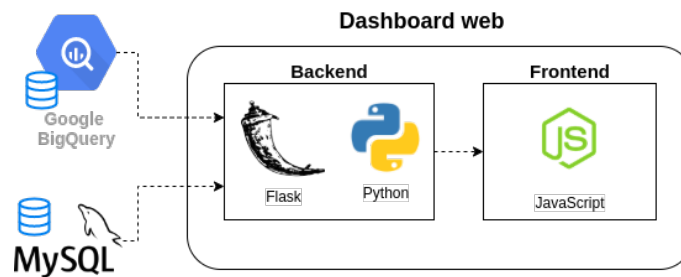


**Figura 3.2:** Flujo de trabajo para detección de demanda constreñida.

### 3.3. Arquitectura

#### 3.3.1. Arquitectura general

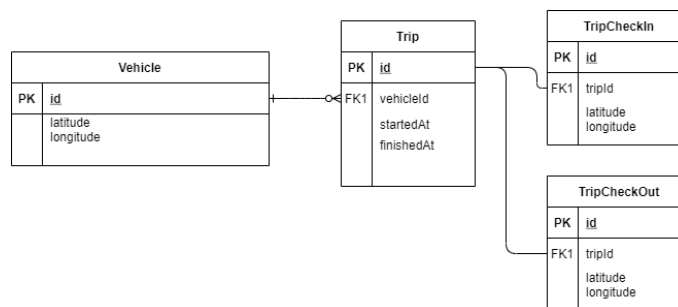
En la figura 3.3 se muestra un diagrama de alto nivel de la arquitectura del *dashboard*. En la sección 2.4 se ha detallado las tecnologías que se usan para los objetivos de este trabajo.



**Figura 3.3:** Diagrama de alto nivel de la arquitectura de la aplicación.

#### 3.3.2. Estructura de la base de datos

En la figura 3.4 se muestra el diagrama entidad relación de la base de datos de MySQL, la tabla *Vehicle* contiene las posiciones de los vehículos en tiempo real. Esta base datos contiene mas campos y tablas que se han obviado.



**Figura 3.4:** Diagrama entidad relación de MySQL.

En la figura 3.5 se muestra los campos de las dos tablas relacionadas con los estados de los vehículos de BigQuery, a diferencia de MySQL no tienen relaciones, *primary keys* ni *foreign keys* ya que son tablas para almacenar registros continuos de los vehículos, aunque sí pueden ser cruzadas entre si mediante el campo IMEI. En el apéndice B se describe cada campo de estas tablas.

Vehiculos	VeBackOffice
IMEI	ID
GPS.TIME	PLATE
GPS.LAT	IMEI
GPS.LNG	STATE
RH.STF	AVAILABLE
CB.VOL	DATETIME

**Figura 3.5:** Campos de las tablas de BigQuery.

## 3.4. Requisitos

La mayoría de los requisitos funcionales y no funcionales son establecidos de acuerdo a la implementación anterior de la aplicación.

### 3.4.1. Requisitos funcionales

- RF-1.**— El desarrollo debe ser en python debido a que el backend es Flask.
- RF-2.**— Se deberá desarrollar la aplicación en un entorno local o de desarrollo para mantener el de producción operativo.
- RF-3.**— El despliegue en el entorno de test y producción se hará mediante el equipo de *DevOps* de la organización.
- RF-4.**— Se deberá poder seleccionar una región específica para el análisis.
- RF-5.**— Se deberá poder seleccionar el tipo de mapa a utilizar para el análisis.
- RF-6.**— Cada región del mapa de tipo *grid* debe coincidir con el mapa que identifica los grids.
- RF-7.**— Cada región del mapa de tipo *grid* debe ser identificable para poder ver qué zona abarca.

### 3.4.2. Requisitos no funcionales

- RNF-1.**— El desarrollo estará integrado con el repositorio de Git y GitHub de la aplicación.
- RNF-2.**— El despliegue en el servidor de desarrollo se hará mediante Docker.
- RNF-3.**— Las consultas hechas mediante el gestor de Google BigQuery deben ser optimizadas para reducir el coste monetario.
- RNF-4.**— El usuario del *dashboard* podrá seleccionar el rango de fechas para el agregado de los datos.
- RNF-5.**— El tiempo de espera para el procesado de información será inferior a 2 minutos.
- RNF-6.**— El desarrollo será en sistema Linux para igualar los entornos de test y producción, también en Linux.
- RNF-7.**— Los colores de las regiones del mapa de tipo *grid* deben contrastar lo suficiente para diferenciar distintos valores



# IMPLEMENTACIÓN

---

En este capítulo se detalla la implementación de las soluciones propuestas en el capítulo anterior así como sus resultados mas significativos.

## 4.1. Cálculo y visualización de la demanda mediante mapas

### 4.1.1. Mapa de estimación de demanda basada en viajes y vehículos

#### Filtro de días por semana

Debido a que la actividad del servicio es diferente dependiendo del día de la semana, un primer paso es implementar un selector en el *frontend* para que se agrupe el número de viajes de acuerdo a los días de la semana elegidos. Al generar el mapa, se hará una consulta a la base de datos para que extraiga el número de viajes realizados sólo los días de la semana elegidos.

#### Afrontar valores a 0

Las ecuaciones 3.1 y 3.2 pueden tener los nominadores a 0: en un determinado grid  $g$  puede haber habido 0 viajes o puede que haya 0 vehículos. Por otro lado, puede ocurrir que haya denominadores a 0 aunque esto es improbable (en toda la ciudad elegida nunca habrá 0 viajes o 0 vehículos)

Debido a que los parámetros anteriores pueden resultar a 0. La ecuación 3.3 puede tener los siguientes problemas:

- 1.– **División 0/0:** en una cuadrícula puede haber habido 0 viajes y haber 0 vehículos.
- 2.– **División nominador>0 y denominador=0:** en una cuadrícula puede haber habido viajes pero no hay vehículos.

Estos problemas se resuelven mediante el código 4.1, que rellena con 0 aquellas zonas que tengan los parámetros de viajes y vehículos de las ecuaciones 3.1 y 3.2 como división entre 0 y se filtran

aquellos *grids* que no tengan ni viajes ni vehículos. Sin embargo la ecuación 3.3 todavía puede tener su denominador a 0, posteriormente se detalla como se gestiona esto.

**Código 4.1:** Filtrado de valores con denominadores a 0 y zonas de no interés

```

1  # se pone a 0 los casos de +inf
2  df_demanda["motos_desplegadas"] = df_demanda["motos_desplegadas"].fillna(0)
3  df_demanda["viajes"] = df_demanda["viajes"].fillna(0)
4
5  # se elimina las zonas con 0 viajes y 0 vehículos (no es de interés)
6  df_demanda = df_demanda[(df_demanda["motos_desplegadas"] > 0) | (df_demanda["viajes"] > 0)]

```

## Optimización de contraste de colores

Para favorecer el contraste de los colores del mapa, se aplica una función logarítmica al valor resultante de la ecuación 3.3, ahora los problemas mencionados con valores a 0 pueden resultar en lo siguiente:

- $\log(\frac{XT_g}{0V_g})$ : resultando un valor de infinito.
- $\log(\frac{0T_g}{XV_g})$ : resultando un valor de menos infinito.

Esto supone un problema a la hora de realizar cálculos pero es necesario disponer de esta información porque el valor a infinito indica que no hay vehículos pero ha habido viajes, lo que es un indicador importante de que en esa zona se necesita vehículos para atender la demanda. El valor a menos infinito indica que en esa zona no ha habido viajes pero sí hay vehículos, es una situación que en pocas ocasiones se puede dar.

El código 4.2 realiza la operación de la ecuación 3.3 (campo “count”), aplica la función logarítmica y convierte los valores infinitos a cadenas de caracteres para su correcto procesado en el *frontend*.

A partir de los valores pasados al *frontend*, se crea tres escalas de colores para los tres casos de infinito, número real y menos infinito, en la sección 5.2.1 se muestra el resultado final.

## 4.2. Detección de demanda constreñida

Como se ha explicado antes, las zonas con demanda constreñida son aquellas zonas en las que se produce escasos viajes porque hay escasos vehículos y sin embargo habría mas viajes si hubiera mas vehículos disponibles allí.

### 4.2.1. Selección e identificación de zona a analizar



**Código 4.2:** Aplicación de función logarítmica y conversión de valores infinitos

```

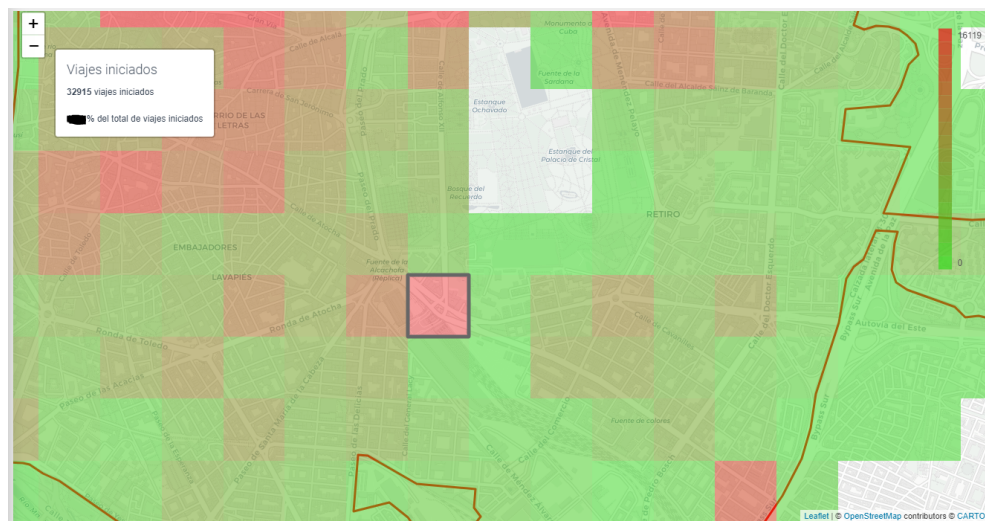
1 df_demanda["count"] = df_demanda["viajes"] / df_demanda["motos_desplegadas"]
2 df_demanda["count_original"] = df_demanda["count"]
3
4 # se aplica escala logarítmica
5 df_demanda["count"] = df_demanda["count"].apply(np.log)
6
7 # inf -> log(xV/0M) -> no hay ninguna moto pero ha habido viajes
8 # -inf -> log(0V/xM) -> hay alguna moto en sitio donde no ha habido viajes (improbable)
9 df_demanda["count"] = df_demanda["count"].apply(
10     lambda num: "inf" if np.isposinf(num) else ("-inf" if np.isneginf(num) else num)
11 )
12 df_demanda["count_original"] = df_demanda["count_original"].apply(
13     lambda num: "inf" if np.isposinf(num) else ("-inf" if np.isneginf(num) else num)
14 )

```

### Selección de la zona

Se ha decidido comenzar este análisis con la zona con mas viajes de Madrid, para ello se ha hecho uso del mapa de tipo *grid* que muestra la cantidad de viajes por cuadrícula. Para tener la información más completa posible, es necesario calcular este dato con todos los viajes históricos de la ciudad, por lo que se aplica el filtro desde el inicio del servicio.

En la figura 4.1 se muestra el *grid* con más viajes de Madrid, se trata de la estación de Atocha.

**Figura 4.1:** Cuadrícula de Madrid con mas viajes realizados, coincidente con la estación de Atocha.

### Identificación de la zona

Para poder realizar consultas y operaciones en la cuadrícula elegida es necesario identificar ese *grid* para disponer de sus parámetros y poder ubicarlo sobre el mapa. El mapa generado por el dash-

*board* de la figura 4.1 no provee de información geográfica de los *grids* mostrados ni seleccionados por lo que es necesario generar el mapa de la cuadrícula junto con un identificador por cada *grid* y que contenga información geográfica de sus vértices.

El código fuente del *dashboard* contiene los vértices del cuadrado que engloba completamente a cada una de las regiones disponibles, este cuadrado se denomina *bounding box*. Cuando se genera un *grid* como el de la figura 4.1, se sigue los siguientes pasos:

- 1.— Se crea cuadrículas dentro de este *bounding box* y sobre toda la región de la ciudad.
- 2.— Se filtran aquellas que no contengan información, por ejemplo, volviendo a la figura 4.1, las cuadrículas del retiro han sido filtradas porque no contienen viajes.
- 3.— Se aplica color acuerdo a su valor en las cuadrículas resultantes.

### Obtención de la información de la cuadrícula

Para disponer de los datos geográficos de la cuadrícula elegida y de otras, se probaron dos soluciones, de las cuales una no funcionó debido a su ineficiencia, esta prueba no exitosa se detalla en el apartado 5.1.2 de pruebas.

La solución llevada a cabo consiste en generar la cuadrícula de la región completa tal y como el paso 1 del *dashboard* y posteriormente se filtra aquellos *grids* que no estén contenidos o toquen la *geofence*. En el código 4.3 el *dataframe* *df* contiene cada cuadrícula junto su geometría (vertices) y se le aplica una columna *inside*. Con una función *lambda*, se calcula mediante *GeoPandas* si cada una de ellas está dentro de la *geofence*. Posteriormente se mantiene aquellas con valor a *inside* a *True* y se reinicia el índice, este índice será el que identifique cada cuadrícula.

**Código 4.3:** Cálculo y filtro de cuadrículas dentro de la *geofence*

```

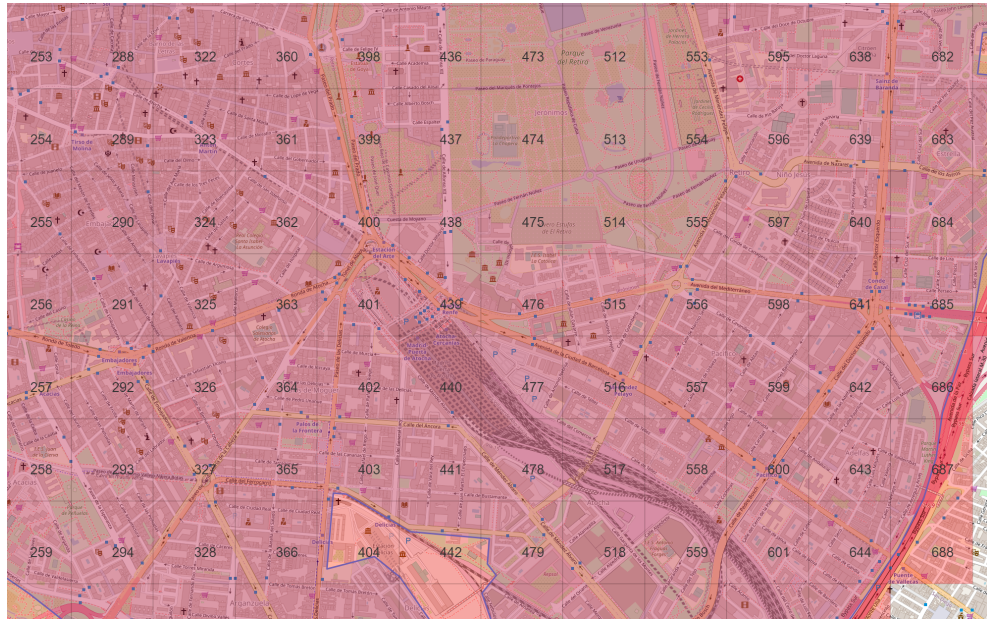
1 df["inside"] = df["geometry"].apply(
2     lambda square: any([square.intersects(poly) for poly in geofence["geometry"]])
3 )
4 df = df[df["inside"] == True]
5
6 df = df.reset_index(drop=True)
7 df["grid_id"] = df.index

```

Haciendo uso de *Folium* se genera un mapa con estas cuadrículas que contienen su identificador. En la figura 4.2 se muestra los *grids* identificados, y se puede ver que el elegido corresponde con el número 439.

Por último, en la misma ejecución se inserta dentro de la base de datos de *MySQL* el *DataFrame* que contiene las cuadrículas con información de su ID y sus vértices mediante *df.to\_sql*.

Con esto ya es posible acceder la posición de sus vértices mediante la base de datos haciendo una consulta al *grid* número 439 de Madrid, el resultado es como el de la tabla 4.1 que contiene las



**Figura 4.2:** Cuadrícula de Madrid con los *grids* identificados, la cuadrícula correspondiente con Atocha es la número 439.

longitudes y latitudes mínimas y máximas que forma ese cuadrado.

city_id	grid_id	min_lon	max_lon	min_lat	max_lat
1	439	-3.6908	-3.6872	40.4057	40.4084

**Tabla 4.1:** Resultado tras consultar la cuadrícula número 439 de Madrid (city\_id = 1)

## 4.2.2. Extracción y unificación de los datos

### Obtención de viajes

El siguiente paso es la obtención de los viajes salientes de la cuadrícula elegida, esto sería un indicador de la demanda en esa zona. También es útil obtener los viajes entrantes a esa cuadrícula ya que sería un indicador adicional a la oferta, es decir, cuando finaliza un viaje allí, el vehículo queda disponible para ser usado otra vez.

En los códigos 4.4 y 4.5 se muestra las consultas de los viaje que empiezan y finalizan dentro del *grid* de Atocha respectivamente.

### Creación de tabla calendario

Antes de unificar los datos, es necesario la creación de una tabla calendario, que tenga como único campo una fecha y hora. La primera fecha corresponderá a la fecha de inicio del servicio y se creará registros cada hora. Este calendario sirve para hacer un `LEFT JOIN` de los vehículos y los viajes, es

**Código 4.4:** Consulta de viajes salientes en Atocha, las condiciones de latitud y longitud son sacadas de la consulta del grid 439 (tabla 4.1)

```

1 SELECT t.started_at as fecha, tc.latitude, tc.longitude
2 from motit.Trip t
3 join motit.TripCheckin tc
4 on t.id = tc.id
5 where tc.latitude >= 40.4057 and tc.longitude >= -3.6908
6 and tc.latitude <= 40.4084 and tc.longitude <= -3.6872

```

**Código 4.5:** Consulta de viajes salientes en Atocha, las condiciones de latitud y longitud son sacadas de la consulta del grid 439 (tabla 4.1)

```

1 SELECT t.finished_at as fecha, tc.latitude, tc.longitude
2 from motit.Trip t
3 join motit.TripCheckout tc
4 on t.id = tc.id
5 where tc.latitude >= 40.4057 and tc.longitude >= -3.6908
6 and tc.latitude <= 40.4084 and tc.longitude <= -3.6872

```

decir, para que quede registrados aquellas horas en las que no hay ni vehículos ni viajes. Si no fuera por esta tabla, habría “huecos” temporales en los que había 0 viajes y 0 vehículos. En la tabla 4.2 se muestra el resultado.

fecha
2018-10-18 04:00:00.0
2018-10-18 05:00:00.0
2018-10-18 06:00:00.0
...
2021-06-01 23:00:00.0

**Tabla 4.2:** Tabla calendario resultante

### Obtención de vehículos disponibles

Para obtener los vehículos disponibles, es decir, que pueden ser usados por los clientes para iniciar viajes, es necesario consultar a BigQuery aquellos vehículos que han estado dentro del *grid* de Atocha usando los campos GPS.LAT y GPS.LNG junto con las siguientes condiciones:

- `STATE="active"`: el campo `active` determina si el vehículo está disponible (no está en revisión, desactivada por avería...)
- `int(RH.STF) % 2 = 0`: `RH.STF` contiene información a nivel de bit sobre el estado de la motocicleta, el primer bit estará a 1 si el vehículo está en viaje, si no está en viaje será 0. Para comprobar que este bit está a 0 simplemente es necesario comprobar que el número es par.

- `CB.VOL >= 48.5`: el campo `CB.VOL` indica el voltaje de la batería. Se considera que la batería tiene suficiente voltaje para estar disponible si es mayor o igual a 48.5 voltios.

En el apéndice B.2 se muestra la consulta completa.

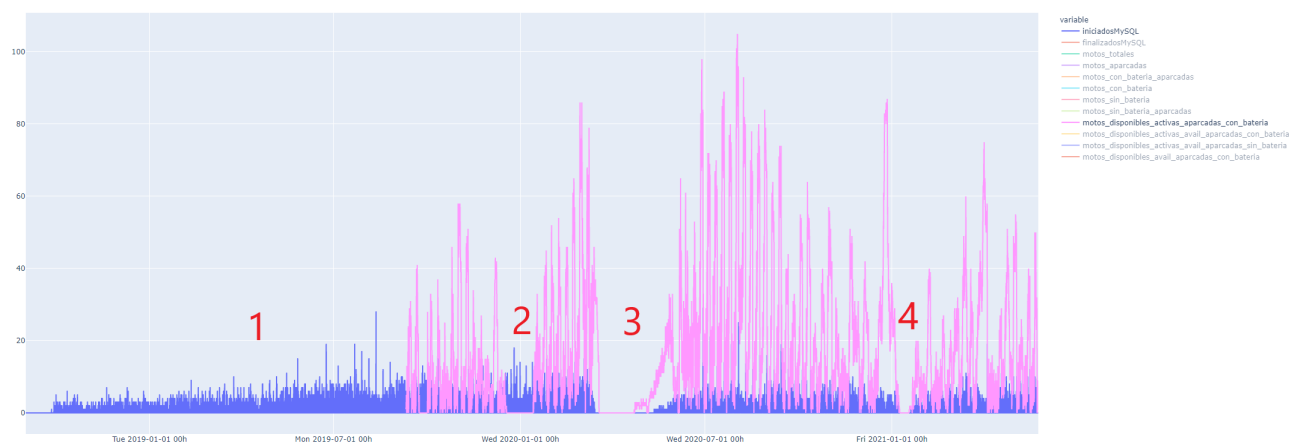
También se ha realizado la obtención de vehículos disponibles usando otros campos, el convenio por el que un vehículo está disponible al cliente ha ido cambiando a lo largo del tiempo por lo que se ha probado a obtener la información usando condiciones de otros campos, estas pruebas se muestran en la sección 5.1.2.

### Unificación de los datos y representación

Los datos de los viajes y vehículos son unidos mediante el campo en común “fecha” que ha sido truncado a la hora en punto del registro. Esto se ha conseguido cargando en un DataFrame `df_bigquery` los vehículos junto con el calendario y en otro DataFrame `df_mysql` los viajes. La operación para unificar ambos DataFrames es

```
df_final = df_mysql.merge(df_bigquery,
how="left", left_on="fecha", right_on="fecha")
```

El resultado se muestra en la figura 4.3, aunque se dispone de información de diferente disponibilidad de vehículos como se explica en 5.1.2, tan solo se muestra los gráficos de “iniciadosMySQL” en azul y “motos\_disponibles\_activas\_aparcadas\_con\_bateria” en rosa. Además, se enumera sobre la figura los momentos en los que los datos tienen comportamiento anormal para ser explicado a continuación.



**Figura 4.3:** Serie temporal completa de viajes y vehículos disponibles de Atocha, en azul son los viajes iniciados y en rosa los vehículos disponibles. Los números rojos son los momentos marcados de inconsistencias de datos.

### 4.2.3. Calidad de los datos

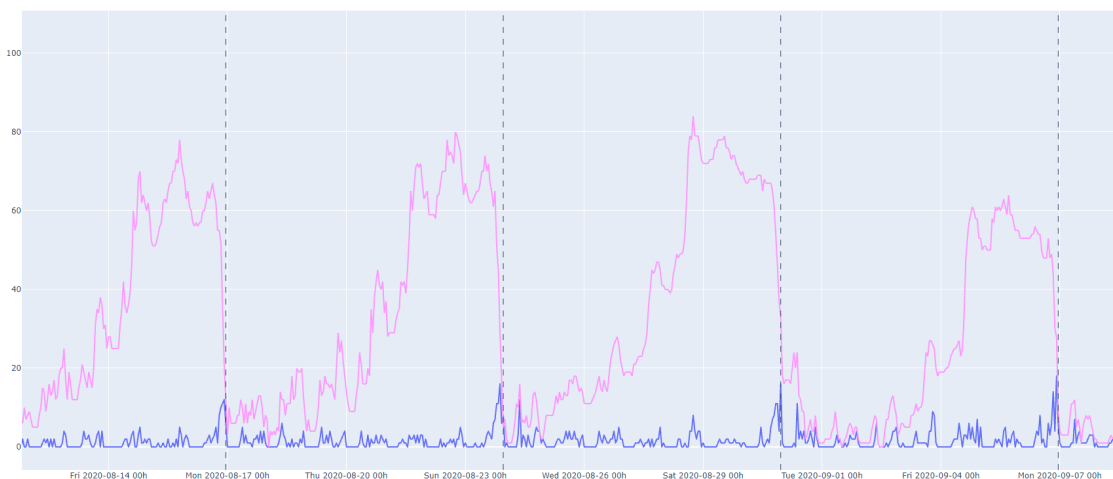
La serie temporal de la figura 4.3 comienza en octubre de 2018 y finaliza en mayo de 2021 y se pueden ver varias caídas de datos que son explicados a continuación.

- 1.— Primera parte de la serie en la que no hay datos de motos disponibles. Esto se debe a que antes de septiembre de 2018 no se registraba los estados de los vehículos aparcados.
- 2.— Caída de BigQuery. La base de datos de BigQuery estuvo caída un mes en diciembre de 2019.
- 3.— Confinamiento por Covid-19. Desde marzo de 2020 hasta mayo el confinamiento por el virus hizo cerrar el servicio.
- 4.— Borrasca Filomena. Esta borrasca en todo Madrid dejó inoperativo el servicio desde el 7 de enero hasta el 21 de enero de 2021.

En la sección 5.1.2 de pruebas se verifica que el número de viajes debe ser menor al número de vehículos disponibles, esto es así en toda la serie excepto en dos periodos aislados por lo que no supone un problema.

### 4.2.4. Detección de patrón

Con un análisis más de cerca se puede observar un patrón interesante: se realizan picos de viajes al rededor de las 21h-23h todos los domingos, haciendo caer la disponibilidad de vehículos al mínimo en muchas ocasiones. En la figura 4.4 se expone un ejemplo con este patrón en 4 domingos seguidos, la línea vertical punteada marca cada domingo a las 23h.



**Figura 4.4:** Patrón de pico de viajes en Atocha, en azul los viajes iniciados y en rosa los vehículos disponibles. La línea vertical punteada marca los domingos a las 23h

Posteriormente se sacó la conclusión de que este patrón no sólo era propio de los domingos, si no de **aquellos días finales de días no lectivos**, en el apéndice C se muestra algunas pruebas de esto, por ejemplo el 11 de octubre de 2020, que era domingo, no se produce este fenómeno debido a que el



día siguiente era festivo. Sin embargo este 12 de octubre (lunes) sí se produce los picos de viajes.

Estos picos de viajes a esas horas hacen caer la disponibilidad de los vehículos y seguidamente cae también el número de viajes por lo que parece ser casos de demanda constreñida.

#### 4.2.5. Predicción y creación de modelo

A partir de la información sacada, se toma como objetivo la predicción y estimación de demanda constreñida en el *grid* de Atocha los domingos alrededor de las 22h-23h.

##### Selección de los datos

Para ello se comienza seleccionando los vehículos disponibles y viajes de sólo aquellos días en los que sucede este fenómeno: los días que marcan el final de días no lectivos. Para esto se sigue los siguientes pasos:

- 1.– **Seleccionar todos los domingos:** la mayoría de días que ocurre esto son los domingos al ser fin de semana (final de días no lectivos).
- 2.– **Añadir días festivos que continúen fines de semana:** tal y como se ha visto, los picos de viajes se desplazan a estos días.
- 3.– **Eliminar los domingos que estén dentro de puente:** cuando un lunes es festivo, los picos de viajes no se producen en domingo.

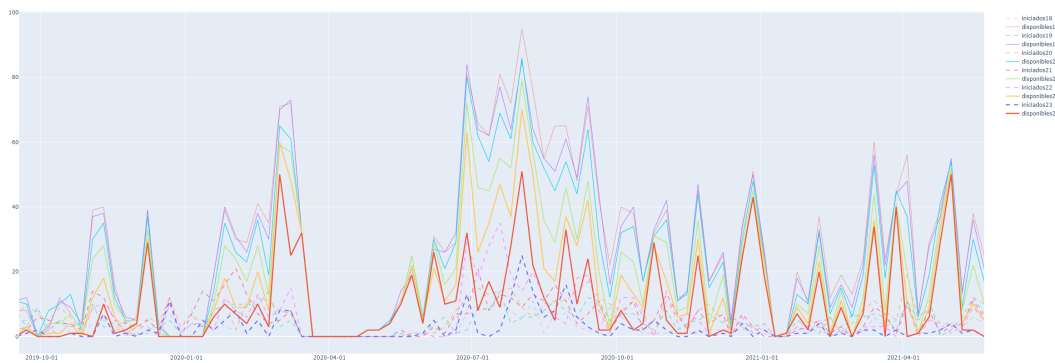
Una vez obtenidos los días. Es necesario obtener las horas a las que se produce el fenómeno: los picos de viajes no se producen en las primeras horas ni a medio día del domingo, tan solo al final de la tarde. Para ello se ha seleccionado un rango horario desde las 18h hasta las 23h.

En la figura 4.5 se muestra el resultado de graficar como eje X todos los días escogidos frente a los vehículos disponibles y viajes iniciados por cada hora en el rango seleccionado. Las líneas de las horas van aumentando en grosor a medida que aumenta las horas para su mejor distinción, las líneas punteadas marcan los viajes iniciados y las líneas continuas los vehículos disponibles.

##### Creación del modelo

Con estos datos se decide crear un modelo para **estimar o predecir los viajes que hubiera habido o habrá a las 22h si hubiera habido suficientes vehículos**. Se ha decidido esa hora porque es cercana a los picos viajes y es cuando en muchas ocasiones se produce demanda constreñida.

Para ello, es necesario ajustar el modelo cogiendo datos de cuando había suficientes vehículos a la hora que se quiere estimar. Es decir, ajustar el modelo sólo con aquellas situaciones en las que no ha habido demanda constreñida en el rango de horas escogido. Analizando la serie temporal de viajes e iniciados de atocha de los días elegidos, se determina que un umbral aceptable en el que hay viajes



**Figura 4.5:** Gráfico en el que se muestra la cantidad de viajes y vehículos disponibles en Atocha los días seleccionados desde las 18h hasta las 23h, el grosor de las líneas va aumentando con la hora. Cada unidad del eje X corresponde con un día de los elegidos. Las líneas punteadas marcan los viajes iniciados y las líneas continuas los vehículos disponibles.

independientemente de los vehículos (no hay demanda constreñida) es 19 vehículos disponibles, es decir, a partir de 19 vehículos o mas, los días elegidos en el rango de horas 18h a 23h, se produce demanda (viajes) sin limitación de la oferta (vehículos).

Con estas variables es posible realizar una regresión multivariable que determine los posibles viajes iniciados a las 22h a partir de la información de vehículos disponibles y viajes anteriores.

El método de regresión a utilizar es el *Backwards stepwise regression* que permite determinar qué variables son relevantes para la regresión y cuales no, de forma que al comienzo se incluye todas las variables y en cada iteración, se va eliminando aquellas con menor relevancia de una en una [21].

Se realiza mediante la librería `statsmodels` de Python siguiendo los siguientes pasos:

- 1.– **Añadir todas las variable dependientes**
- 2.– **Se realiza *Ordinary Least Squares*** para obtener los parámetros de la regresión.
- 3.– **Ajuste del modelo.** La librería de `statsmodel` devuelve con la función `fit()` los resultados del ajuste.
- 4.– **Eliminar aquella variable con *P-value* mayor y que sea  $p > 0,05$**
- 5.– **Volver al paso 2** si queda alguna variable con  $p > 0,05$ , si no, se detiene el algoritmo y las variables resultantes serán los parámetros finales.

El código 4.6 muestra la primera iteración y en la figura 4.6 se muestra su resultado en la que están incluidas todas las variables inicialmente, tras esto habría que ir eliminando de las variables dependientes aquellas que cuyo *p-value* sea mayor que 0.05.

El resultado final y su interpretación se muestra en la sección 5.2.2.



**Código 4.6:** Ordinary least squares y fit de la primera iteración del *Backwards stepwise regression*

```

1 # dep_cols contiene las variable dependientes o regresores
2 X = df_serie_domingos_plt_cols_th[dep_cols]
3
4 X2 = sm.add_constant(X)
5 est = sm.OLS(y, X2)
6 est2 = est.fit()

```

```

=====
                        OLS Regression Results
=====
Dep. Variable:      iniciados22      R-squared:      0.699
Model:              OLS              Adj. R-squared: 0.572
Method:             Least Squares    F-statistic:    5.519
Date:               Wed, 16 Jun 2021  Prob (F-statistic): 0.00108
Time:               16:45:14         Log-Likelihood: -86.862
No. Observations:   28              AIC:             191.7
Df Residuals:       19              BIC:             203.7
Df Model:           8
Covariance Type:    nonrobust
=====

```

	coef	std err	t	P> t	[0.025	0.975]
const	-6.7720	4.594	-1.474	0.157	-16.387	2.843
iniciados18	1.1300	0.722	1.565	0.134	-0.381	2.641
disponibles18	-0.2469	0.555	-0.445	0.661	-1.409	0.915
iniciados19	0.4726	0.694	0.681	0.504	-0.980	1.925
disponibles19	0.3608	0.475	0.760	0.457	-0.633	1.355
iniciados20	0.3596	0.810	0.444	0.662	-1.335	2.054
disponibles20	0.2746	0.867	0.317	0.755	-1.540	2.090
iniciados21	-0.1732	0.483	-0.358	0.724	-1.185	0.838
disponibles21	-0.2575	0.841	-0.306	0.763	-2.018	1.503

```

=====
Omnibus:            1.535      Durbin-Watson:      1.668
Prob(Omnibus):      0.464      Jarque-Bera (JB):    1.067
Skew:               -0.475      Prob(JB):            0.587
Kurtosis:           2.893      Cond. No.            393.
=====

```

**Figura 4.6:** Primera iteración de *Backwards stepwise regression* en la que están incluidas todas las variables, el siguiente paso sería eliminar la variable “disponibles21” ya que tiene el valor  $P > |t|$  mayor.



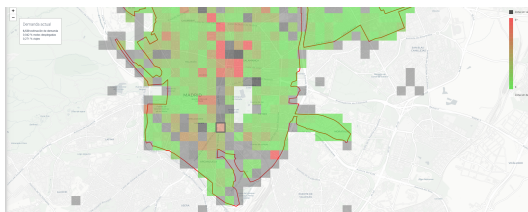
## PRUEBAS Y RESULTADOS

### 5.1. Pruebas

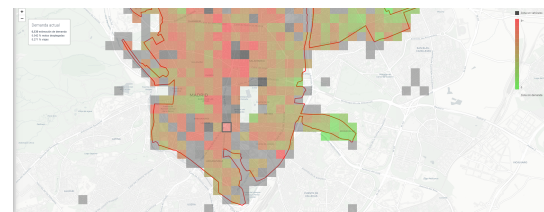
#### 5.1.1. Pruebas realizadas para la visualización con mapa

##### Pruebas para la optimización de contraste de colores

Como se ha explicado en la sección 4.1.1 en la que se implementa el mapa de estimación de demanda, se ha aplicado una escala logarítmica para visualizar mejor los valores de cada cuadrícula, en la figura 5.1(a) y 5.1(b) muestran el resultado antes y después de aplicar la función logarítmica al parámetro de estimación de demanda.



(a) Contraste de colores sin aplicar la función logarítmica



(b) Contraste de colores tras aplicar la función logarítmica

**Figura 5.1:** Resultado de aplicar escala logarítmica, la figura 5.1(a) muestra las cuadrículas coloreadas con el valor original y la figura 5.1(b) tiene aplicado la función logarítmica

#### 5.1.2. Pruebas realizadas para la detección de demanda constreñida

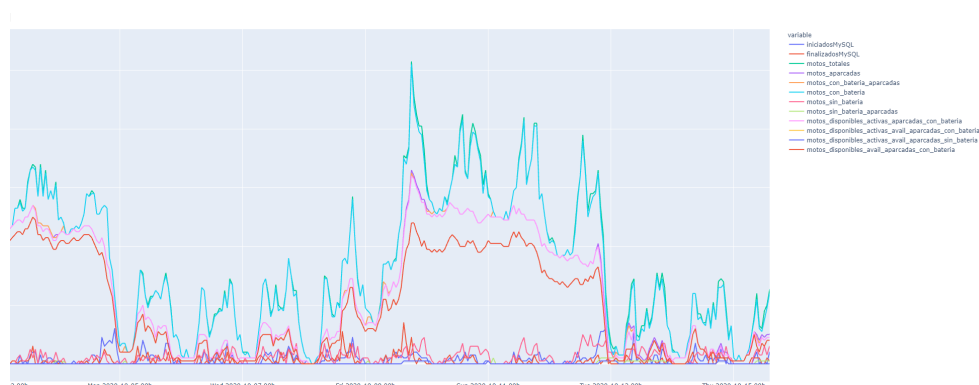
##### Obtención de la cuadrícula junto con su información geográfica

Tal y como se ha mencionado anteriormente, una prueba que se hizo para generar la identificación de los *grids*, era extraer todos los viajes del histórico de la ciudad elegida, posteriormente calcular todos aquellos viajes que estuvieran dentro de la *geofence* y generar toda la cuadrícula a partir de esto.

El problema era la operación de calcular los viajes dentro de la *geofence* porque el coste computacional de realizar operaciones geoespaciales de cientos de miles de viajes con las diferentes *geofences* de Madrid (compuesta por las zonas de Alcobendas, Pozuelo y Madrid) es inmenso. Se lanzó este proceso en la máquina de desarrollo de Google Cloud pero tras dejarla inoperativa durante varias horas, se decidió abordar el problema de otra forma.

## Obtención de vehículos con diferentes condiciones

En la figura 5.2 se muestra la serie temporal mostrando todas las gráficas de viajes y vehículos del *grid* de Atocha dependiendo de diferentes condiciones de las motocicletas, esto incluye estados de los vehículos como aparcados, aparcados sin batería etc.



**Figura 5.2:** Sección del gráfico con todos los datos de viajes y vehículos de Atocha

## Comprobación de la consistencia de los datos

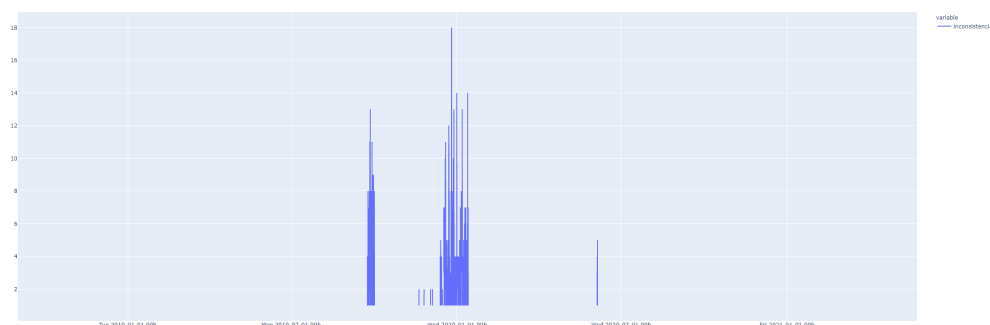
Esta comprobación trata de verificar que la serie temporal de Atocha de viajes iniciados y vehículos disponibles es consistente. Esto se hace mediante la resta de *vviajes* – *disponibles*, si el resultado es negativo, entonces será una hora con inconsistencia de datos.

En la figura 5.3 se muestra los periodos (cada unidad del eje X es una hora) en el que los viajes son mayores que los vehículos disponibles, la dimensión del eje Y indica el grado de inconsistencia, es decir *vviajes* – *disponibles* cuando viajes es mayor que disponibles. Tan solo se produce en periodos aislados por lo que no supone un problema.

## Análisis de los periodos con demanda constreñida

Como se ha mencionado en la implementación, en la cuadrícula de Atocha, los domingos a las 23h se produce picos de viajes que hacen caer la oferta de vehículos al mínimo siendo un indicador de demanda constreñida, es decir, los momentos en los que hay pocos viajes porque hay pocos vehículos.

En la figura 5.4 se muestra el porcentaje de días de la semana en los que se produce que

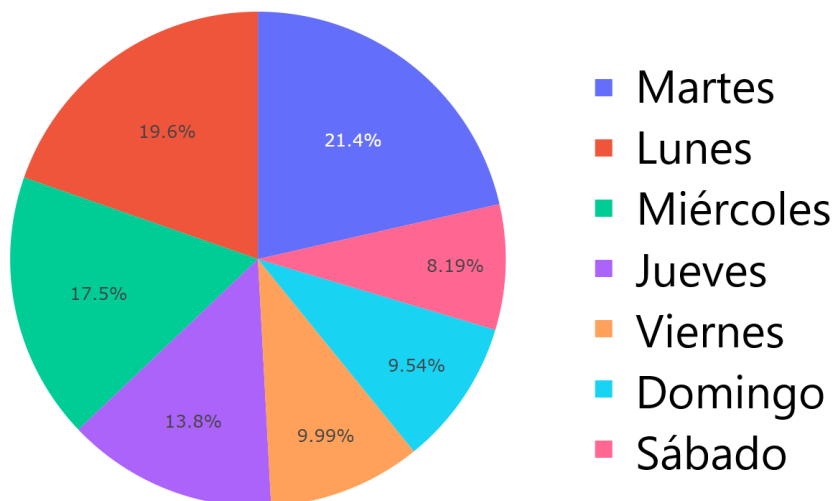


**Figura 5.3:** Gráfico en el que se representa cuándo se produce inconsistencias de viajes y vehículos

$Vehiculos - Viajes \leq T$ , siendo  $T$  un umbral para determinar que hay casi tantos viajes como vehículos, en este caso 3. Según lo visto y este resultado, esto se podría interpretar que, en esta zona, los domingos es cuando se vacía de vehículos disponibles haciendo que los siguientes días haya poca demanda porque hay poca oferta. A medida que continua los días, la zona se va llenando de vehículos (mas viajes entrantes que salientes) hasta que vuelve a vaciarse en domingo.

Un posible problema con esta figura es que se han incluido los datos los datos de inconsistencias, caídas de base de datos y el confinamiento por Covid-19.

Días de la semana con disponibles-iniciados  $\leq 3$



**Figura 5.4:** Porcentaje de días de la semana en los que hay indicador de demanda constreñida  $Vehiculos - Viajes \leq 3$ . El orden de la leyenda está por orden de ocurrencia.

## 5.2. Resultados

### 5.2.1. Resultados de la implementación la visualización con mapa

A continuación se muestra el resultado final de la solución de la estimación de demanda mediante mapa y sus interpretaciones.

Las figuras 5.5(a) y 5.5(b) muestran el resultado final de esta solución, se ha seleccionado los datos de los viajes resultantes de los días viernes, sábado y domingo en Madrid del periodo 17 de abril de 2021 hasta el 6 de junio de 2021.

Sobre las zonas marcadas con números de la figura 5.5(b) se interpreta lo siguiente:

- 1.– Esta cuadrícula está en negro porque no contiene ningún vehículo y es más oscura que otras (como la zona número 2 y cercanas de la derecha que tampoco tienen vehículos) porque en ella se ha realizado más viajes. Al seleccionar este *grid* se tiene la información de que ahí se ha realizado un 0.537 % de viajes. Esto indica que en esta zona es muy posible que haya un déficit de oferta de vehículos.
- 2.– Esta zona presenta el mismo problema que la 1, sin embargo, su color es menos oscuro lo que indica que aunque allí no haya motos, no ha habido tantos viajes, teniendo menos prioridad que la anterior.
- 3.– Zona en verde claro, se trata de una zona en la que se encuentran vehículos y ha habido viajes, la importancia de la demanda en esta zona depende de los datos obtenidos al seleccionarla, además, las cuadrículas adyacentes en rojo, indican que hay más necesidad de disponibilidad de vehículos, ya sea porque ha habido más demanda o hay menos oferta.

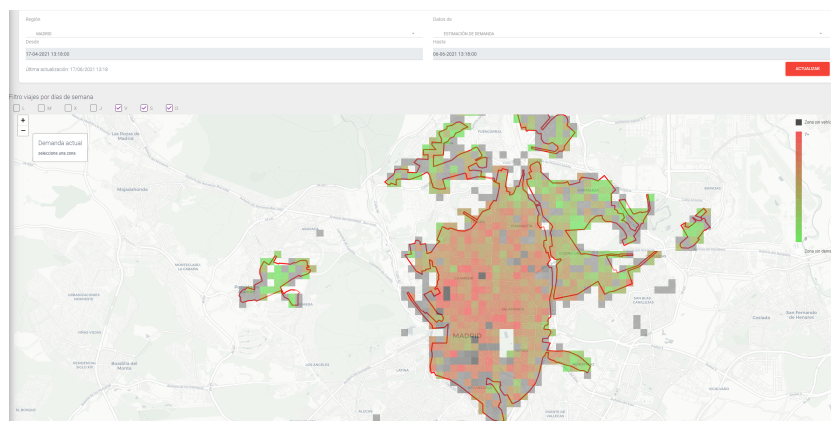
### 5.2.2. Resultados la detección y predicción de la demanda constreñida

El resultado de realizar los pasos completos del *Backwards stepwise regression* que se aplica en la sección 4.2.5, se muestra en la figura 5.6. Esto indica que las variables finales relevantes se tratan del número de viajes iniciados a las 18h y el número de viajes iniciados a las 20h con un coeficiente de 1.055 y 1.066 respectivamente. Estos coeficientes indican que ambas variables tiene prácticamente el mismo peso a la hora de estimar la variable dependiente.

En la figura 5.7 se muestra el valor real a predecir (viajes iniciados a las 22h) y el valor predicho del modelo sólo cuando hay suficientes vehículos a las 22h o en otras palabras cuando la demanda es independiente de la oferta a esa hora.

Esta predicción permite estimar cuantos viaje se hubieran hecho a las 22h los finales de días no lectivos (que se trata de momentos de gran demanda) si hubiera habido suficiente oferta, además de predecir los valores de estos viajes dado el número de viajes iniciados a las 18h y el número de viajes iniciados a las 20h.

La precisión del modelo está condicionada por los datos de la serie temporal: los periodos de indisponibilidad de vehículos, los días de lluvia que hacen caer notablemente la demanda y el toque de queda establecido por el gobierno por la pandemia hace que la precisión del modelo no sea del todo exacta. Esto también hace tener escasos datos para aplicar otros modelos de series temporales como



(a) Visualización general de Madrid de estimación de demanda



(b) Visualización mas detallada, se ha seleccionado una cuadrícula en la que no hay vehículos actualmente

**Figura 5.5:** Resultado del mapa de estimación de demanda basado en viajes y vehículos, la figura muestra el contexto general y la figura se ve mas en detalle, sobre esta se ha puesto 3 marcadores.

OLS Regression Results						
=====						
Dep. Variable:	iniciados22	R-squared:	0.657			
Model:	OLS	Adj. R-squared:	0.629			
Method:	Least Squares	F-statistic:	23.91			
Date:	Wed, 16 Jun 2021	Prob (F-statistic):	1.57e-06			
Time:	16:45:15	Log-likelihood:	-88.708			
No. Observations:	28	AIC:	183.4			
Df Residuals:	25	BIC:	187.4			
Df Model:	2					
Covariance Type:	nonrobust					
=====						
	coef	std err	t	P> t	[0.025	0.975]
-----						
const	-2.6805	2.178	-1.231	0.230	-7.166	1.805
iniciados18	1.0555	0.425	2.486	0.020	0.181	1.936
iniciados20	1.0665	0.279	3.827	0.001	0.493	1.646
-----						
Omnibus:	0.788	Durbin-Watson:	1.733			
Prob(Omnibus):	0.674	Jarque-Bera (JB):	0.178			
Skew:	-0.168	Prob(JB):	0.915			
Kurtosis:	3.200	Cond. No.	19.8			

**Figura 5.6:** Última iteración de *Backwards stepwise regression*, las variable resultantes son iniciados18 e iniciados20.

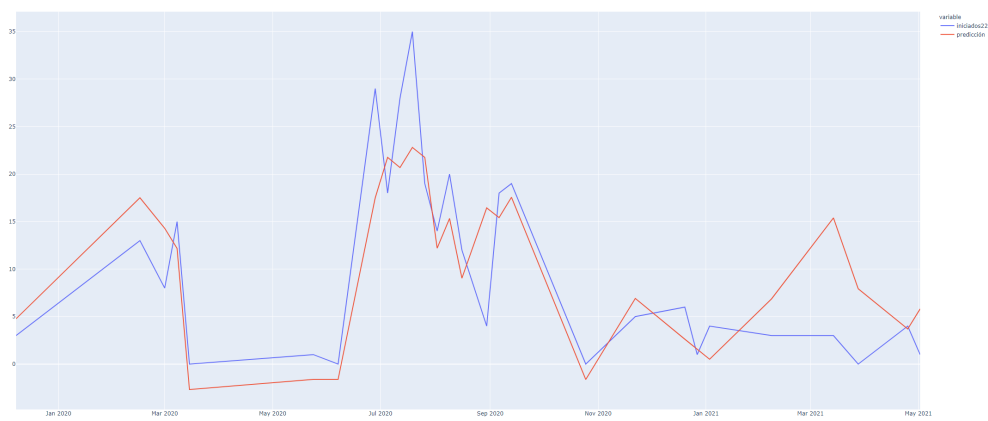


Figura 5.7: Predicción del modelo (línea roja) y valor real (línea azul).

los explicados en la sección 2.2 del estado del arte.



## CONCLUSIÓN Y TRABAJO FUTURO

---

### 6.1. Conclusión

Los servicios de movilidad compartida que permiten a sus clientes empezar y finalizar viajes en cualquier sitio una la zona permitida pueden crear problemas de desbalance de flota, haciendo que en zonas de alta demanda haya escasa oferta de vehículos o en contrapartida, zonas con baja demanda se encuentren demasiados vehículos.

Esto supone un problema para los clientes del servicio debido a que disponer de un vehículo cercano cuando este quiera desplazarse es clave para que ese usuario pueda realizar el viaje. También supone un problema para el operador del servicio: en las grandes ciudades existen numerosas empresas competidoras que tienen sus flotas repartidas por la ciudad haciendo que el cliente pueda elegir realizar el viaje con aquella empresa que tenga más disponibilidad cercana de vehículos.

En este trabajo se ha creado un recurso de visualización de zonas en las que indican déficit o superávit de vehículos de acuerdo a su registro de viajes pasados.

También se ha descubierto, como caso particular, que en la estación de Atocha, la zona con más actividad del negocio, se produce demanda constreñida, es decir, que a una hora determinada de alta demanda están faltando vehículos y de haberlos se producirían más viajes para atender la demanda real de los clientes. Se ha creado un modelo que permite estimar cual sería el valor real de viaje realizados si hubiera habido suficientes vehículos en esa zona así como predecir el número de viajes que se harán en esa zona a partir de otros valores pasados.

Disponer de información sobre aquellas zonas en las que se necesite tener más vehículos y sobre aquellas zonas en las que se necesite retirar es clave para optimizar el problema de desbalance de oferta de vehículos. Este trabajo permite mejorar la toma de decisiones operativas a la hora de reubicar vehículos en aquellas zonas que se necesite.

## 6.2. Trabajo futuro

En el futuro es posible complementar los resultados de este trabajo de forma que la solución de estimación de la demanda mediante el mapa de cuadrículas se aplique para establecer precios dinámicos a los clientes en función de la necesidad de disponer o retirar vehículos: aquellas zonas en las que haya baja disponibilidad de oferta sería una zona de precio reducido para finalizar el viaje, de igual forma, aquellas zonas con demasiados vehículos tendrían precio reducido para iniciar viajes. De esta forma se reduciría los gastos operativos.

Respecto a la detección y predicción de demanda constreñida, se podría trabajar para obtener mejores y más datos ya sea sin situaciones extremas de por medio como el Covid-19 u obtener datos de meteorología. Se podría generalizar y detectar todas aquellas zonas de las ciudades operativas en las que está ocurriendo esto. Automatizando el proceso para todas las cuadrículas también sería posible saber cuándo pasará y anticipar en esa zona reubicación de vehículos.

# BIBLIOGRAFÍA

---

- [1] “Parques de vehículos, tablas estadísticas 2019.” <https://www.dgt.es/es/seguridad-vial/estadisticas-e-indicadores/parque-vehiculos/tablas-estadisticas/2019/>. Accessed: 2021-04-25.
- [2] R. Nair, E. Miller-Hooks, R. C. Hampshire, and A. Bušić, “Large-scale vehicle sharing systems: analysis of vélib’,” *International Journal of Sustainable Transportation*, vol. 7, no. 1, pp. 85–106, 2013.
- [3] S. Shaheen, N. Chan, A. Bansal, and A. Cohen, “Shared mobility: Definitions, industry developments, and early understanding,” *Transportation Sustainability Research Center, Innovative Mobility Research*, 2015.
- [4] R. Nair and E. Miller-Hooks, “Fleet management for vehicle sharing operations,” *Transportation Science*, vol. 45, no. 4, pp. 524–540, 2011.
- [5] M. Ampudia-Renuncio, B. Guirao, R. Molina-Sánchez, and C. E. de Alvarez, “Understanding the spatial distribution of free-floating carsharing in cities: Analysis of the new madrid experience through a web-based platform,” *Cities*, vol. 98, p. 102593, 2020.
- [6] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.
- [7] J. H. Jones, “Time series and spectral analysis.” <http://web.stanford.edu/class/earthsys214/notes/series.html>. Accessed: 2021-05-24.
- [8] “Overview of autocorrelation, seasonality and stationarity in time series data.” <https://analyticsindiamag.com/an-overview-of-autocorrelation-seasonality-and-stat>. Accessed: 2021-05-25.
- [9] K. C. Clarke, “Advances in geographic information systems,” *Computers, environment and urban systems*, vol. 10, no. 3-4, pp. 175–184, 1986.
- [10] V. Maliene, V. Grigonis, V. Palevičius, and S. Griffiths, “Geographic information system: Old principles with new capabilities,” *Urban Design International*, vol. 16, no. 1, pp. 1–6, 2011.
- [11] J. Snow, *On the mode of communication of cholera*. John Churchill, 1855.
- [12] S. W. A. Gunn, M. Masellis, and S. W. A. Gunn, *Concepts and practice of humanitarian medicine*. Springer, 2008.
- [13] J. M. Last, *A dictionary of public health*. Oxford University Press, USA, 2007.
- [14] “Standard representation of geographic point location by coordinates.” <https://www.iso.org/standard/39242.html>. Accessed: 2021-05-29.
- [15] “seaborn.heatmap.” <https://seaborn.pydata.org/generated/seaborn.heatmap.html>. Accessed: 2021-05-29.

- [16] "A tiny, simple and fast heatmap plugin for leaflet." <http://leaflet.github.io/Leaflet.heat/demo/>. Accessed: 2021-05-29.
- [17] "Interactive choropleth map." <https://leafletjs.com/examples/choropleth/>. Accessed: 2021-05-29.
- [18] "Bigquery." <https://cloud.google.com/bigquery>. Accessed: 2021-05-29.
- [19] "Leaflet." <https://leafletjs.com/>. Accessed: 2021-06-01.
- [20] "Statsmodels api." <https://www.statsmodels.org/stable/index.html>. Accessed: 2021-06-03.
- [21] H. S. Wilf and A. Ralston, *Mathematical methods for digital computers*. John Wiley & Sons, 1960.

# APÉNDICES



# ENTORNOS

---

Este trabajo se realiza en diferentes entornos para su desarrollo, de pruebas y productivo.

- **Entorno local:** es el dispositivo en el que se realiza la implementación, en este se comprueba que se ejecuta o compila correctamente el progreso.
- **Entorno de desarrollo:** servidor, en este caso, de Google Cloud en el que se despliega las primeras implementaciones funcionales, tiene pocos recursos computacionales para ahorrar costes.
- **Entorno de pruebas:** otro servidor en el mismo proveedor con la finalidad de hacer pruebas intensivas a los desarrollos que se piensa tener completados
- **Entorno de producción:** entorno en el que se despliega la implementación final, tiene más recursos computacionales para mantenerse operativo al completo.





# BASES DE DATOS

---

## B.1. Descripción de los campos de las tablas de BigQuery

Las dos tablas mencionadas de BigQuery se compone de los siguientes campos:

- **Tabla Vehículos**

- IMEI: número de identificación del módulo de conexión del vehículo.
- GPS.TIME: tiempo del módulo del GPS.
- GPS.LAT: latitud.
- GPS.LNG: longitud.
- RH.STF: información a nivel de bit del estado del vehículo.
- CB.VOL: voltaje de la batería.

- **Tabla VeBackOffice**

- ID: número de identificación del vehículo, correspondiente al ID de MySQL
- PLATE: matrícula.
- IMEI: igual que en "Vehículos".
- STATE: estado de disponibilidad de la moto.
- AVAILABLE: estado adicional de disponibilidad.
- DATETIME: fecha.

## B.2. Consultas para la generación de datos para las series temporales

El código B.1 muestra la consulta que obtiene el número de vehículos activos en la región de Atocha

**Código B.1:** Consulta para obtener los vehículos disponibles en Atocha

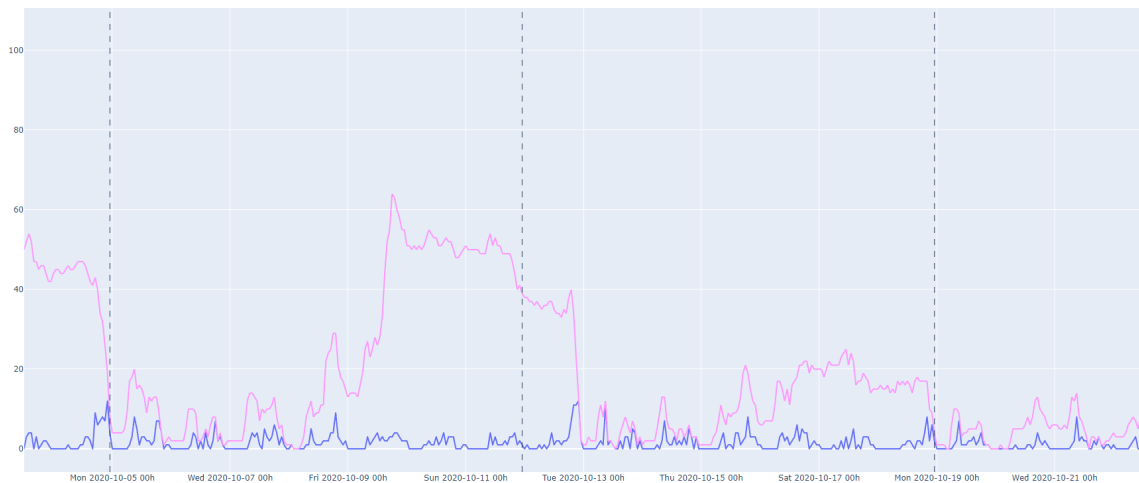
```
1  create table `dlinnovacion.motos.serie_motos_disponibles_activas_aparcadas_con_bateria` as
2  select bo.fecha, count(distinct ve.imei) as motos_disponibles_activas_aparcadas_con_bateria
3  from
4  (
5  select e.fecha, e.imei, e.tiempo_estado_actual, a.available, a.state
6  from
7  (
8  select f.fecha, v.imei, max(v.datetime) as tiempo_estado_actual
9  from `dlinnovacion.motos.calendario_ok` as f
10 left join `motit-motosharing.DL_Innovacion.VeBackoffice` v on f.fecha>=v.datetime
11 where v.imei>1
12 group by f.fecha, v.imei
13 --order by f.fecha
14 )e
15 join `motit-motosharing.DL_Innovacion.VeBackoffice` a
16 on e.imei=a.imei and e.tiempo_estado_actual=a.datetime
17 --order by e.fecha
18 )bo left join `motit-motosharing.DL_Innovacion.Vehiculos` ve
19 on bo.imei=ve.imei and bo.state='active' and mod(cast(ve.rh.stf as int64),2)=0 and cb.vol>=48.5 and
    ve.gps.lat>=40.405746098265894 and ve.gps.lat<=40.40846721387283 and ve.gps.lng>=-3.69081584
    and ve.gps.lng<=-3.68726588 and DATETIME_TRUNC(ve.gps.time, HOUR)=bo.fecha
20 group by bo.fecha
21 order by bo.fecha
```

# PATRONES DE VIAJES EN FINALES DE LECTIVOS

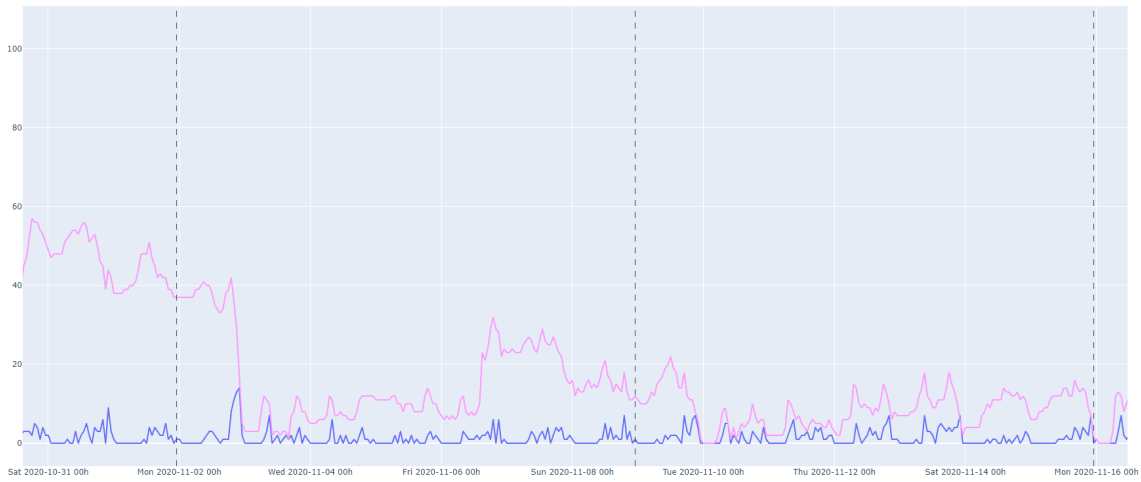
---

## C.1. Patrones de picos de viajes en finales de días no lectivos

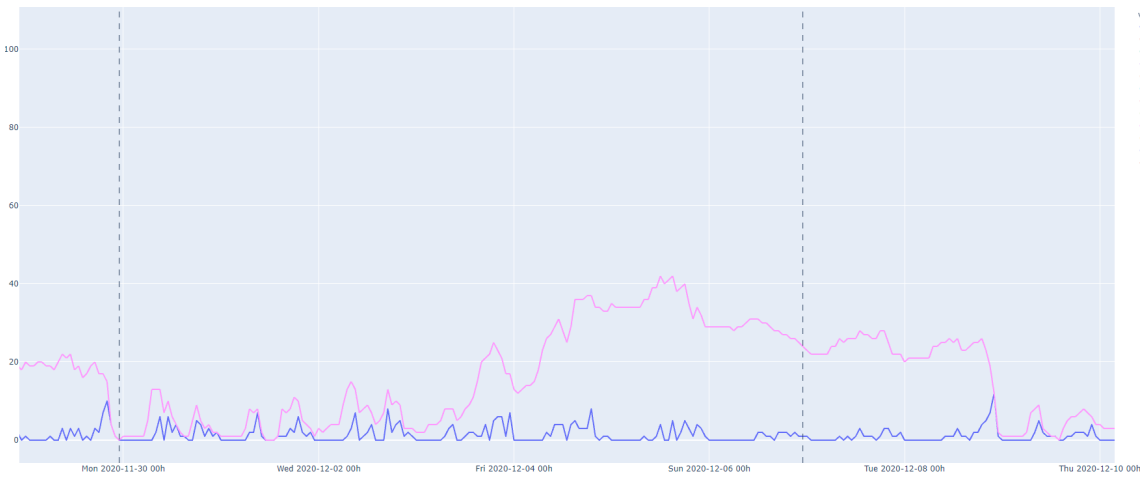
En las figuras C.1, C.2 y C.3 se muestran como los patrones de picos de viajes se desplazan a aquellos días que son finales de puente.



**Figura C.1:** Pico de viaje desplazado del domingo 11 al lunes 12 de octubre de de 2020. La línea vertical central marca el domingo 11 de octubre, sin embargo, no se produce es epico de viajes debido a que el día siguiente es fiesta



**Figura C.2:** Picos de viajes desplazados por festivos el 2 Lunes y 9 Lunes de Noviembre de 2020 (dos primeras líneas verticales)



**Figura C.3:** Pico de viajes desplazado al 9 de diciembre de 2020. En este caso se juntaron dos días festivos seguidos y sin embargo el pico de viaje sigue desplazándose hasta el día que finaliza el puente



